Design, Development and Evaluation of a Big Data Analytics Dashboard

Master's Thesis

Copenhagen
Business School
HANDELSHØJSKOLEN

M.Sc. Business Administration and Information Systems
(Information Management)

International Study Programme (2014)
Double Degree
Copenhagen Business School & University of Mannheim

Author: Benjamin Flesch
CPR-Nr.:

submission date: 4. Sept 2014
120139 characters, 72 pages

Signature: ______________________

Supervisor: Prof. Dr. Ravi Vatrapu
Department of IT Management
Copenhagen Business School

Preface

First of all, I want to thank Prof. Ravi Vatrapu for supervising me during the writing of this thesis and for supporting me during my research activities besides his comprehensive duties as professor at Copenhagen Business School and his work at the Computational Social Science Laboratory.

Furthermore, I would like to mention Sophia and Aaron as they kindly supported me by reading various drafts of this thesis.

Abstract

This Master's thesis focuses on the Design, Development and Evaluation of a novel Visual Analytics Dashboard for Big Data Analytics. The presented dashboard connects social activity from Facebook with a thorough event timeline of the factory disasters in the Bangladesh garment industry. Bangladesh depicts one of the largest garment industries in the world, and their mostly female workers only receive a low wage.

The goal of this thesis is to present a thorough understanding of the design and development processes needed to implement a Big Data Visual Analytics tool based on freely available open-source components in a robust, extensible manner. Moreover, an evaluation of the developed dashboard is performed based on a task-based user study in conjunction with software and database performance optimization. The user study concludes that the dashboard is easy to use in a productive manner without prior training and experience in using visual analytics tools.

By using the presented dashboard, even novice users can gain profound understanding of the tragedies in Bangladesh, their background, and the resulting social media impact. Furthermore, the linked social media activity from eleven international companies in the garment industry can be interactively explored through different visualizations depicting actor mobility, conversation content, language distribution, and overall activity levels.

Contents

List of Figures

List of Tables

List of Listings

1. Introduction

1.1. Motivation

In recent years, visual analytics tools have steadily been improved and adapted in order to work with large data sets, so-called Big Data, while providing accessibility to a growing audience. Although many of these data sets have historically been of proprietary nature, the growth of social media also spurred availability of huge collections of social media activity. Various social networks, such as Twitter and Facebook, provide extensive data while being increasingly used for research and business purposes.

In light of current research on social media activity in regard to major events, such as the 2013 German Bundestag elections using Facebook and Twitter data (Kaczmirek et al.; 2013), further analysis of distinctive events and their social media impact needs to be performed using state of the art visual analytics tools.

Under these preconditions, the 2012/2013 Bangladesh textile industry disasters which prompted massive exclamation from consumers and media outlets all over the world present a series of events worth studying on a global scale. Both the viewpoints of consumers and high-profile textile industry brands alike are captured in the social media dataset, with the latter publicly perceived as an adversary to workers' rights in the textile industry.

Given the opportunity to source social media data from Facebook for a perfectly sufficient timeframe for analysis of the Bangladesh textile industry event timeline, an interactive visual analytics dashboard based on the available data is designed and developed in context of this thesis.

To assess the quality of the developed dashboard, its accessibility and interactive components are evaluated by means of user and software testing. The evaluation is performed in order to gain a thorough quantitative and qualitative reasoning in regard to the value created for end users when visually analyzing complex event timelines such as the above-mentioned factory disasters in the textile industry. Additional insight on the impact of these events on social media activity and waves of sentiment against specific social media actors can be obtained through this analysis.

1.2. Goal

The purpose of this thesis is to review the design, development and evaluation processes of a visual analytics dashboard which is intended to process massive social media data sets, so-called Big Social Data, side on side with complex event timelines.

In the scope of this Master's thesis, I intend to contribute by presenting a novel approach on the design and implementation of a visual analytics dashboard for the analysis of complex event timelines which are linked to bulk, semi-structured social media data from Facebook.

The architectural design of the visual analytics dashboard aims at laying a sound foundation for future abstractions to a general-purpose social big data event timeline visualization tool in anticipation of processing larger amounts of data in the future.

On top of the technically challenging design and development of the IT artifact at hand, a thorough evaluation using a range of different methodologies will be performed with the goal of presenting actionable insights on the design of interactive and big data-ready visual analytics dashboards.

During all these efforts, a particular focus lies on responsiveness of the dashboard to the end user in terms of speed and ease of use, and the adaptability to varying devices and screen sizes with form factors ranging from handheld devices to high-resolution, wall mounted displays.

1.3. Structure

The structure of this thesis is as follows:

The first two chapters give insight on the underlying research questions and the technical background; the next chapter describes the topic of acquiring and processing the social data set and accompanying event data; whereas chapters four and five concern the design and development of the IT artifact which represents the visual analytics dashboard.

In chapter six, a thorough evaluation of the implementation results is performed and evaluation results are presented. The final three chapters concern analysis of related work in this field of research, a summary of findings and the conclusion.

2. Theoretical Background

This chapter introduces background information on the topics of the ongoing factory disasters in the Bangladesh garment industry which spiked in 2012/2013 and their direct and indirect impact on the textile industry and their public perception. It then illuminates the history and underlying concepts of Visual Analytics, Big Data and implications that arise from the combination of both in regard to social media data and complex event timelines.

2.1. Bangladesh Factory Disasters

The garment industry in Bangladesh is the second-largest exporter of clothing after China, and employs more than 3 million - mainly female - workers. This is emphasized by Bajaj (2012, p.2) in reference to a large factory fire in Bangladesh at the 25th of November 2012 which killed 112 workers.

The garment industry in Bangladesh has rapidly grown during the past 20 years while approving of lax safety regulations and frequent accidents (Sato; 2014). "Bangladesh's garment sector [..] employs forty percent of industrial workers and earns eighty percent of export revenue. Yet the majority of workers are women. They earn among the lowest wages in the world and work in appalling conditions. Trade unions and associations face brutal conditions as labour regulations are openly flouted" (Khanna; 2011, p.1).

At April 24th, 2013, factory disasters in the Bangladeshi garment sector culminated in the largest textile industry tragedy to date with the collapse of *Rana Plaza*, a factory building in an industrial suburb of Bangladesh's capital Dhaka (Manik et al.; 2013), in which more than 1100 garment workers died during the factory's collapse and subsequent fires (Burke; 2013). This event has been reported by media outlets all over the world and deeply shocked many end consumers of clothing products originating from Bangladesh.

In various research publications, safety and struggles of workers in the Bangladesh garment industry have been widely discussed (Khanna; 2011), also with special regard to ongoing protests (Himi and Rahman; 2013), globalization-related problems (Rahman; 2013) and ethical aspects of the factory disasters (Stewart; 2013).

Nevertheless, the lack of publicly shown empathy by many major textile industry companies created a public outcry against perceived unethical behavior in textile industry supply chains. In many cases, this public outcry was expressed by consumers and directly addressed to the respective clothing brands, which were in the consumers' immediate reach through means of social media channels such as Facebook.

The factory disasters in Bangladesh prompted major consumer-facing textile industry brands like *H&M* and *Walmart* to join campaigns supporting textile workers' rights in Bangladesh. A more sustainable, but lagging impact is felt by the introduction of better methods of supply chain management such as social contracts in supply chains (Islam et al.; 2014).

2.2. Visual Analytics

Visual analytics is defined as "the science of analytical reasoning facilitated by interactive visual interfaces" (Thomas and Cook; 2005, p.4). The scientific notion was first presented by Wong and Thomas (2004).

"Visual analytics aims at making data and information processing transparent," and enables a constructive analysis where "humans and machines cooperate using their respective, distinct capabilities for the most effective results [..], while the user has ultimate authority in directing the analysis" (Keim et al.; 2010, p.2).

Scientific use of Visual Analytics

Tools with Visual Analytics functionalities are applied in different fields of study for the exploratory analysis of complex relationships between non-trivial quantities of data.

Biomedial Research In biomedical research, bioinformatics presents us with a range of Visual Analytics tools that support the analysis of complex diseases. This includes the “BiNA” tool for graph-based analysis of biological network data (Gerasch et al.; 2014) and the “Hawkeye” tool for interactive Visual Analytics in genome assembly analysis (Schatz et al.; 2007). Further tools facilitate graph-based network analysis for research on diseases like Alzheimer’s and asthma, where “visual analytics [presents a] considerable overlap with the fields of scientific visualization, and information visualization” (Bhavnani et al.; 2013, p.2).

Investigative Research Analysts’ investigative research activities greatly benefit from Visual Analytics tools. This is showcased by a study on simulated intelligence analysis tasks where different investigative strategies are applied using the “Jigsaw” visual analytics tool, with the four major strategies being ‘**Find a Clue, Follow the Trail**’ (FCFT), ‘**Hit the Keyword**’ (HTK), ‘**Build from Detail**’ (BFD) and ‘**Overview, Filter, Detail**’ (OFD). Most Visual Analytics tools allow their users to choose between multiple visualization methods based on the needs of their current investigative strategy. The Jigsaw tool in particular presents eight different kinds of visualization only for the intelligence analysis of documents (Kang et al.; 2011).

Forensic Accounting & Forensic Analysis Visual Analytics tools are also used in forensic accounting for uncovering fraudulent transactions and transaction schemes. Although posing a major benefit in forensic investigations, Visual Analytics are not leveraged to their fullest potential as there is a particular dependency on confidential business records and financial data from proprietary databases, for which the data acquisition processes are complicated and time-consuming.

In these forensic activities, Visual Analytics are often reduced to graph-based visualizations, which showcase the connections between entities. A notable example depicts the historical forensic analysis based on public business records of 18th century business networks the shipping industry (Haggerty and Haggerty; 2009).

Business Intelligence For business intelligence purposes, a wide variety of Visual Analytics tools exist. Besides renowned proprietary products such as Tableau (Tableau Software Inc; 2014), SAS (SAS Institute Inc.; 2014) and Spotfire (TIBCO Software Inc; 2014), the research community has created Visual Analytics tools such as "VizDeck", a general-purpose, on-the-fly dashboard creation tool for "exploratory visual analytics of unorganized relational data, [... which] automatically recommends appropriate visualizations based on the statistical properties of the data" (Key et al.; 2012, p.1).

Social Media Data We will have a thorough look at Visual Analytics tools particularly designed for the analysis of social media data in section 2.4.

2.3. Big Data

Big Data is a modern terminology for large data sets which are hard to process with traditional tools due to their sheer size. Although the concept encompasses new technological developments in database and storage technology, it remains mainly a marketing term for interdisciplinary processing of available information. According to Buhl et al. (2013, p.66), "above all, [Big Data] is a multidisciplinary and evolutionary fusion of new technologies in combination with new dimensions in data storage and processing (volume and velocity), a new era of data source variety and the challenge of managing data quality adequately (veracity)".

Big Social Data is a term for Big Data which is obtained from the social media world. With Facebook and Twitter being some of the the most popular social networks on the internet, the data they provide to third parties over public and private APIs (such as the Twitter Firehose) can be called Big Social Data.

2.4. Visual Analytics of Big Social Data

As this thesis presents an IT artifact visualizing the Bangladesh factory disaster event timeline in light of social media activity from various companies' virtual

Facebook presences, a particular blend of technologies in Visual Analytics and Big Social Data is crucial.

Wong et al. (2012) underline major challenges that arise in Visual Analytics of data sets classified as Big Data, in particular scalability, summarization, difficulties in achieving smooth interactive UIs, and the development of effective methods for user-driven data reduction.

The analysis of social media actors, their actions and the artifacts they create is actively performed by researchers and businesses with focus on many different topics. For one, researchers use advanced methods of sentiment analysis to gain insight into the "reaction of people to events, topics and entities" based on Twitter data (Bravo-Marquez et al.; 2014, p.2). Additional commercial, web-based tools such as SAS exist which have a specialization on exploratory Visual Analytics of Big Data (Abousalh-Neto and Kazgan; 2012); but they lack the complexity needed for working with combined Big Social and event timeline data.

To conclude the theoretical background, previous research presents a healthy assessment of opportunities that arise through the use of Visual Analytics on large data sets, but on the other hand underlines several problems in handling of Big Social Data through traditional means of Visual Analytics software which need to be accounted for during implementation of the IT artifact.

3. Data Sources and Acquisition

This chapter introduces the two main data sources which are used in development of the Big Data Analytics dashboard. Data from these two equally important data sources is used during the realization of the Visual Analytics dashboard depicted in the first introductory chapter:

The first data source consists of social media activity from the virtual online presences on Facebook - so-called “Facebook walls” in domain language - of a wide selection of retail companies in the garment industry.

The second data source consists of a timeline of various events in regard to the publication of the series of Bangladesh factory disasters in years 2009 to 2014 via consumer-facing news outlets.

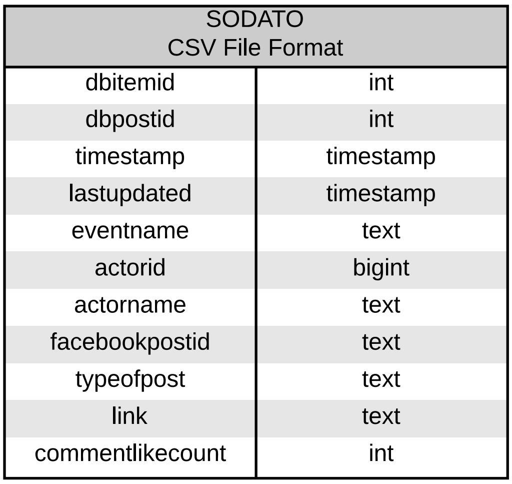
3.1. Social Media Activity from Facebook

Social media activity from Facebook is acquired using the Social Data Analytics Tool (SODATO) presented by Hussain and Vatrapu (2014) for eleven major companies in the clothing retail industry.

Only companies that present retail stores in Europe and/or the United States are selected in order to gain a representative set of social media activity that contains consumers' reaction to the Bangladesh disasters from above mentioned geographical regions. The final selection of companies from the retail clothing industry is as follows:

1. Benetton
2. Calvin Klein
3. Carrefour
4. El Corte Ingles
5. H&M
6. JC Penny
7. Mango
8. Primark
9. PVH
10. Walmart
11. Zara

After successful data extraction using the Facebook API, the SODATO tool supplies the Facebook activity for each company as a .CSV file which follows a specific data format convention.



SODATO CSV File Format	
dbitemid	int
dbpostid	int
timestamp	timestamp
lastupdated	timestamp
eventname	text
actorid	bigint
actorname	text
facebookpostid	text
typeofpost	text
link	text
commentlikecount	int

Figure 3.1.: Structure of SODATO-supplied Facebook data sets

Figure 3.1 showcases the data format of the SODATO output. Each piece of Facebook activity is uniquely identified by the tuple of (**dbitemid**, **dbpostid**). Both **timestamp** and **lastupdated** contain timestamp information.

The most important distinction between different social media activity types depicts the **eventname** field, which holds any one of the values *[POST, COMMENT, LIKE]*. The **eventname** field thereby depicts the Action performed by the social media Actor who is specified by **actorid** and **actorname**.

The field **facebookpostid** is the unique identifier of each Facebook conversation, whereas **typeofpost** specifies the exact type of the Artifact as one of *[status, SWF*[1]*, photo, offer, music, link, video, question]*. The field **link** is optional and in some cases contains a hyperlink that will be shown in the Facebook UI, but it is of no relevance for this thesis. Finally, the **commentlikecount** field depicts the number of likes which were performed by other Actors on the Artifact at hand.

[1] Shockwave Flash file format is an interactive media format heavily used in creating websites and web browser based games until 2010.

Data Collection In early June 2014, Facebook activity within the period from January 2009 to June 2014 was collected using the SODATO tool for the eleven companies listed above.

Table 3.1.: Overview of Social Media Activity in the Clothing Retail Sector collected from Facebook

Data Source	# Posts	# Comments	# Likes	Total
Benetton	2.411	51.156	3.760.914	3.814.481
Calvin Klein	12.390	44.224	3.196.564	3.253.178
Carrefour	3.711	18.651	79.855	102.217
El Corte Ingles	21.211	121.684	3.168.950	3.311.845
H&M	100.461	262.588	7.779.411	8.142.460
JC Penny	24.744	154.620	3.064.581	3.243.945
Mango	3.498	204.695	18.661.291	18.869.484
Primark	1.343	73.229	1.333.181	1.407.753
PVH	66	80	1.801	1.947
Walmart	284.523	2.147.994	44.812.653	47.245.170
Zara	3.136	12.437	246.294	261.867
Total:	**457.494**	**3.091.358**	**86.105.495**	**89.654.347**

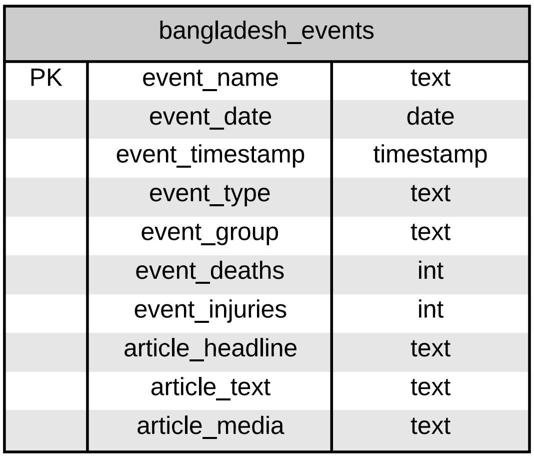
Table 3.1 illustrates extent and diversity of the collected social activity data from Facebook. More than 89 million artifacts from eleven different Facebook walls in the clothing retail space were collected for analysis purposes, with the goal of being able to get a representative subset of consumers of mainstream retail clothing chains.

From the data overview it is apparent that companies' total social media activity varies widely, even though many of them are of comparable size in terms of in-store customers and revenue. This phenomenon might be the result of differing focus on social media marketing by the leadership of the respective organizations.

Overall, we observe with particular interest that US-headquartered Walmart has the most social media activity on Facebook over the whole time period, representing more than half of the total amount of data at our disposal.

3.2. Timeline of Bangladesh Textile Industry Events

To generate the event timeline, an analysis of news stories related to Bangladesh garment factory disasters, workers' rights in Bangladesh and political consequences of the tragedies has been performed. All these events related to the Bangladesh textile industry disasters have been collected manually and used to build a thorough event timeline, with the objective of visualizing the impact of garment factory disasters on Facebook activity of consumer-facing clothing retail companies. The event timeline contains detailed information about the severity of Bangladesh-related incidents and gives numbers of injuries, deaths and other metrics.



bangladesh_events		
PK	event_name	text
	event_date	date
	event_timestamp	timestamp
	event_type	text
	event_group	text
	event_deaths	int
	event_injuries	int
	article_headline	text
	article_text	text
	article_media	text

Figure 3.2.: Structure of manually created Bangladesh Textile Industry Event Timeline

Figure 3.2 illustrates the data format of the Bangladesh event timeline. The field **event_name** depicts the unique name of the event, whereas **event_date** and **event_timestamp** give exact date and timestamp information. The field **event_type** holds any one of the values *[Factory Incident, Protest, Political, Business Impact]* and depicts a broad classification of the event at hand. The fields **event_deaths** and **event_injuries** depict the number of deaths and injuries which are reported by the media source for this event.

The final three fields **article_headline**, **article_text** and **article_media** give further information on the event and reference the underlying article and the media outlet where it was published.

3.3. Data Processing Steps

Various data processing steps need to be performed before the social data and the event timeline can be used for visual analytics purposes. These processing steps are a major component of the data acquisition pipeline.

Processing of Event Timeline Due to the manual collection of all information which is included in the event timeline, no further processing steps apart from conversion of the spreadsheet into .CSV file format is needed.

Processing of Facebook data The SODATO-provided Facebook activity data sets are generated as independent files for each company's Facebook wall, and need to be combined into one for using them as a whole data set that can be filtered or expanded on demand.

Figure 3.3 illustrates the data acquisition process through which the author was able to obtain social media data from Facebook for a wide range of international clothing retailers.

The general concept follows the stages of the "Big Data Value Chain" introduced by Miller and Mork (2013), with steps of preparation, organization and integration of the data prior to visualization and analysis.

Data preparation tasks are performed in a pre-processing step which converts all .CSV files to from their character encoding *UTF-16* to the more commonly used *UTF-8* and handles edge cases in which the generated SODATO output lacks proper data type encapsulation.

Subsequently, a data normalization phase performs sanity checks on the input data and identifies malformed data or unneeded information.

Lastly, all distinct data sets are aggregated while conserving information regarding their original source in an additional variable. The aggregated data is then imported into a database management system (DBMS), from which it can be accessed for visual analytics purposes.

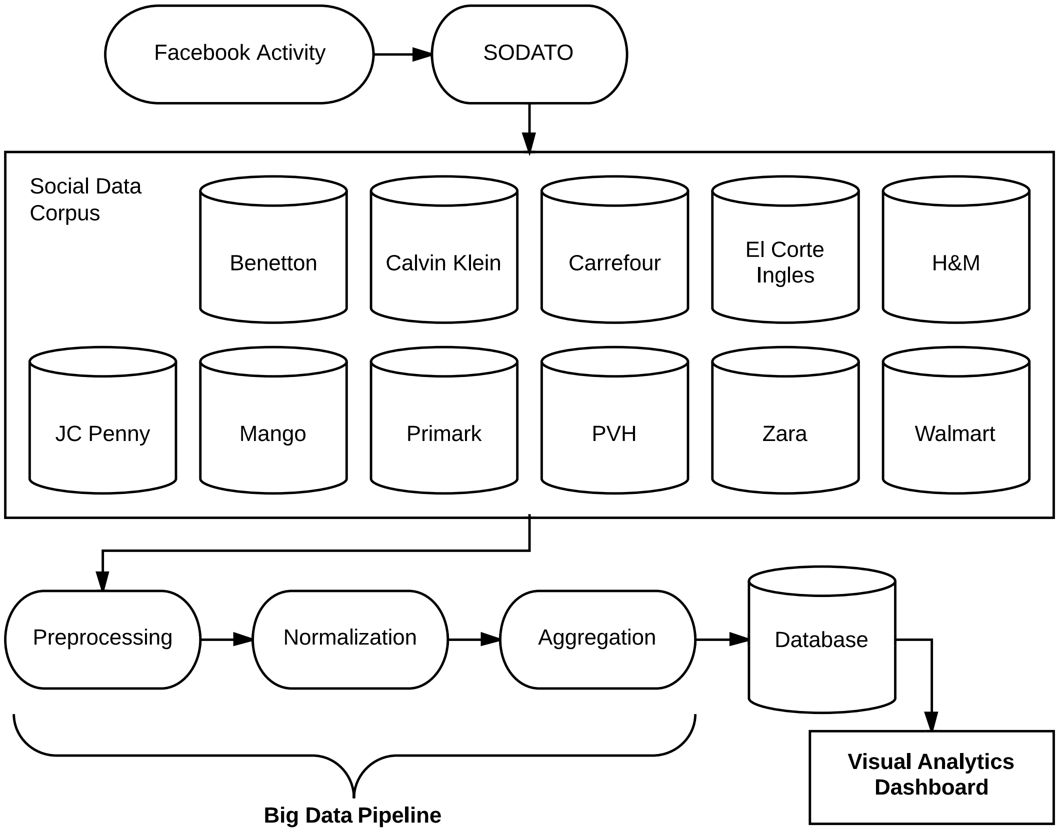


Figure 3.3.: Big Data Acquisition Pipeline of the Social Data from Facebook used later on in the Visual Analytics Tool

3.4. Data Storage

The character of the two underlying data sets which are used in the future visual analytics dashboard has been outlined in previous sections. This section discusses the storage of the acquired data, which will be used later on for development of the dashboard.

Storage of Event Timeline The structured nature of the Bangladesh event timeline makes it easy to choose a fitting storage type for the events. Based on this, a relational database management system (RDBMS) has been chosen. The decision is further reinforced by the rather small amount of events (96) and the simple data structure without external dependencies.

Storage of Facebook Activity Data The semi-structured nature of the Facebook activity data which was acquired from the Facebook walls of eleven companies using the SODATO tool and processed according to the steps described in section 3.3 presents a more difficult decision.

With a total of 8GB, the combined file size of the raw Facebook activity data is quite large, but obviously not large enough to warrant the use of large-scale distributed database systems such as *Apache Hadoop*, based on which Google realized the first Map/Reduce-style processing of Big Data (White; 2009).

Choice of DBMS This presents us with the opportunity to employ broadly established open-source RDBMS like *PostgreSQL* instead of distributed Big Data ready databases like *Apache Hadoop*. This is due to the fact that multi-core systems with beefy hardware and plenty of working memory in the double-digit Gigabyte range are easily available nowadays.

For obvious reasons, a strict ordering of all events and social data by their timestamps and dates is needed for the creation for a visual analytics dashboard as outlined in previous chapters. Given the structured analyses which need to be performed asynchronously in an interactive, user-specified manner, the decision was made to use a RDBMS, with the availability of SQL's expressiveness in a RDBMS being a major contributing factor for this choice. Queries stated in SQL allow for easier modeling of requirements than NoSQL and Map/Reduce approaches of distributed database systems, and present no drawbacks in the use cases at hand.

To conclude this chapter, the data storage for the subsequent steps in creation of the visual analytics dashboard will be performed in a single-machine *PostgreSQL* database which can be queried using SQL. After data import, table optimization and index creation, the final size of all data in the database approaches 80 GB. This means the data can be kept in working memory to a good extent, thereby improving overall performance of the database.

4. Visual Analytics Dashboard Design

In this chapter, the design process of the Visual Analytics dashboard is outlined. Furthermore, the fundamental components of the dashboard are chosen based on the available datasets and possible visualizations. Then, different ways of realizing the Visal Analytics dashboard are discussed and a decision is made. Finally, the architecture of the underlying software and database is presented.

4.1. Design Goals & Objectives

The design of a Visual Analytics Dashboard as depicted in the previous chapters needs clear goals in terms of visualization options, interactive components, target devices and many more. During subsequent steps, for example when implementing the dashboard, these previously formulated goals and objectives provide crucial orientation in many decisions and trade-offs that have to be made.

Multidimensionality A Visual Analytics Dashboard consists of a mash-up of multiple visualizations which can be utilized by the user in combination to maximize efficiency. The type and size of each visualization need to be carefully evaluated.

Accessibility The dashboard should be accessible as easily as possible for users. It should therefore have as few hard dependencies in terms of installed software, operating system or device type as possible.

Responsiveness The dashboard needs to be responsive to different device types and screen sizes. The realized IT artifact should be reasonably accessible using devices with different form factors and adapt accordingly. Both a 4K display used in a conference room and a normal tablet should be able to display the dashboard. Consequentially, the dashboard should also work without problems on a standard-sized workstation computer and modern laptops.

Performance Another key objective for the visual analytics dashboard displays performance of both user- and server-facing software components. The large-scale data set which is the foundation of the dashboard created in this thesis necessitates increased processing needs. This also impacts overall memory footprint and disk space consumption. In order to achieve smooth performance in productive use, some sort of duty-sharing between server and client software components needs to be established. Thereby, workload may be shifted as needed and user interface waiting times are reduced.

Ease of Use From the end user's perspective, ease of use depicts an important non-functional requirement which needs to be addressed. Factually, all software is made accessible to end users in combination with additional documentation, a user manual or even training lessons. Even though these measures present additional value for the end user, the visual analytics dashboard should be designed in a way that enables the user to work with the dashboard without any prior briefing or training on how to use it.

Extensibility Lastly, during realization of the visual analytics dashboard, an extensible framework should be used so that future changes can be implemented with only moderate effort and without unnecessary technical hindrances.

4.2. Design Principles

The design of a Visual Analytics dashboard needs to follow a set of core principles, through which the above stated goals can be achieved.

Key outtakes from a thorough investigation on design principles for visual analytics dashboards by Kang et al. (2011) are that dashboards need to facilitate clue-finding, have smooth transitions between different stages of analysis, support evidence marshalling, allow flexibility in organizing and support task resumption after pursuing alternative paths of analysis.

Furthermore, Keim et al. (2010, p.7) propose that the design of a Visual Analytics dashboard should enable the user to “provide timely, defensible and understandable assessments” of the underlying data.

Additional principles applied to design and realization of the dashboard are:

- **Detail on Demand**: The detail on demand principle strives to first present an easily graspable overview to the user and not the full depth of the available data. This overview can be processed visually and intellectually in short time. Only subsequently, when the user decides to, the level of detail shown in the visual analytics tool can be increased.
- **Ready-made Visualizations**: The Visual Analytics dashboard presented in this thesis is based on social media data from Facebook. The dashboard consists of a combination of multiple visualizations. Therefore, each visualization needs to highlight unique features of the underlying social interactions. This allows the dashboard as a whole to be kept clean and organized, and prevent it from becoming too complex.
- **User-centric Design (UCD)**: Abras et al. (2004, p.2) emphasize that in user-centric design, “the role of the designer is to facilitate the task for the user and to make sure that the user is able to make use of the product as intended and with a minimum effort to learn how to use it”. When designing the interface, a focus is put on optimization of the user experience. This goes as far as directly consulting the end user to find out additional requirements. Hence, a strong accentuation is put on the user of the dashboard to seamlessly solve visual analytics tasks.

Important Considerations when designing dashboards for Big Data As previously discussed in section 2.4, important considerations arise when performing visual analytics due to very large data sets.

Research by Wong et al. (2012) indicates that under these circumstances, various problems may occur which are hard to circumnavigate. One problem is that human cognitive capability might not be able to keep up with the growth of data available for visualization purposes, thereby posing a challenge to end users. Another problem depicts user-driven data reduction, in which the design of user interfaces for data reduction becomes increasingly difficult as new dimensions of data are integrated into visual analytics tools.

4.3. Dashboard Components

In this section, essential components of the Visual Analytics dashboard will be discussed. These components can be realized based on the Facebook activity data set in conjunction with the Bangladesh factory disaster event timeline, which were introduced in sections 3.1 and 3.2 respectively. Afterwards, a decision on the implementation of fundamental visualizations which should be included within the dashboard, can be made based on the discussion results.

4.3.1. Main Activity Chart

The most important dataset for the Visual Analytics dashboard depicts the social activity data from Facebook. Therefore, a *Main Activity Chart* is designed which contains an overview over all social media data at hand. The main activity chart also needs to provide extensive user-driven data reduction and detail enhancement functionality, in order to comply with the *detail on demand* design principle.

Therefore, a state-of-the-art realization of visual analytics concepts, as they are already established within other social media visual analytics tools, is required (Diakopoulos et al.; 2010). Based on these characteristics, dashboard users are enabled to thoroughly analyze the presented social data in a productive and time efficient manner.

4.3.2. Visualization of Actor Mobility over Time and Space

Based on the social media activity data from Facebook, visualizations concerning the Actors and their mobility within the whole data corpus can be created. Important aspects depict the mobility of actors within different time frames and between different Facebook walls.

Actor Mobility over different Timeframes This depicts the feature to visualize all actors that use social media to interact with any of the selected Facebook walls within a specific time frame. Additionally, a comparison of actor overlap between different timeframes enables a more thorough event analysis by means of the dashboard. The overlap of actors between different periods of time, such as *Before*, *During* and *After* a specific event, can visually convey important information. Using a visualization of actor mobility, actor churn rate and growth phenomena of the total user base can more easily be shown.

Therefore, actor mobility on Facebook between different timeframes can be expressed as

$$\text{Actors}_{Time}(start, end) := \{a \mid a \in Actors \wedge \ start \leq a.date \leq end\} \quad (4.1)$$

where *start* and *end* describe the time period from which all Actors are received.

Actor Mobility between Facebook Walls This is a crucial feature of the Visual Analytics dashboard, as it provides information about the overlap of social media actors between different companies and textile industry brands. By leveraging a visualization which addresses this challenge, end users can use the dashboard to determine which actors interact with multiple Facebook walls.

Therefore, actor mobility between different Facebook walls can be expressed as

$$\text{Actors}_{Source}(arr\_sources) := \{a \mid a \in Actors \ \wedge \ a.source \in arr\_sources\} \quad (4.2)$$

where *arr_sources* depicts a list of the Facebook walls from which actors are selected.

General purpose Actor Mobility calculations Naturally, an abstraction of the two different methods of time- (4.1) and wall-based (4.2) overlaps between actors presented above can be created. The abstraction takes both the originating

Facebook Wall, specified in *arr_sources*, and the activity timeframe of the actor on the specific Facebook wall(s), specified by *start* and *end*, into account:

$$\text{Actors}\ (arr\_sources, start, end) := \text{Actors}_{Source}(arr\_sources) \cap \text{Actors}_{Time}(start, end) \quad (4.3)$$

This results in a model where actor mobility is described through the intersections between different sets of actors.

Using the presented model, questions such as *"Which Actors interacted with Walmart and H&M Facebook walls both during the time period from 01. Nov 2013 to 01. Dez 2013 and after it?"* can easily be answered:

$$\begin{aligned} Actors_{Question} &= Actors_{During} \cap Actors_{After} \\ &\text{with:} \\ Actors_{During} &= Actors([H\&M, Walmart], 01.11.2013, 01.12.2013) \\ Actors_{After} &= Actors([H\&M, Walmart], 01.12.2013, \infty) \end{aligned}$$

To recap, in this subsection, a model for describing actor mobility has been developed. Based on this model, answering of complex questions with regard to actor mobility is possible. A visualization of actor mobility can be achieved with any set-based visualization method, for example a proportional Venn diagram.

4.3.3. Visualization of Conversation Content

The overview over the collected Facebook activity in section 3.1 highlighted the availability of many social media artifacts which are classified as Facebook *POSTs* and *COMMENTs*. These artifacts present user-provided content within their *textvalue* attribute, which is displayed in the native Facebook UI as part of a conversation between different Actors. This conversation data can be used in the Visual Analytics dashboard by means of various methods of analysis and visualization.

Sentiment Analysis & Visualization Different approaches to sentiment analysis can be used to automatically classify user-submitted conversation content on a sentiment scale, e.g. using a continuous scale ranging from *happy* to *sad*. The visualization would then present user sentiment for each Facebook wall or for specific time periods.

The downside of sentiment analysis is that the classification algorithms need to be specifically chosen and modified for each data source that is processed. In the present case, the classifier needs to be trained with a sufficiently large percentage of the overall Facebook activity data corpus in order to minimize false positive classifications.

This presents a major technical and financial challenge in regard to multi-million social media artifacts which would have been used for training the classifier. On top of this challenge, further problems arise. The user sentiments expressed in the Facebook activity at hand are often influenced by macro-scale events which might not even be seen in relation to events of the Bangladesh event timeline.

In addition, preliminary manual analysis of conversation content hinted at specific transaction patterns between the textile industry brands which manage the Facebook walls and their customers who reach out via social media. The majority of these transactions can be described as either support requests - in which the consumer writes on the wall and asks for help with regard to a specific product or shopping experience - or consumer reactions to news and pictures published by the textile industry brand. These transaction patterns in the conversations at hand made sentiment analysis not appear a worthwhile effort.

Furthermore, discouraging results with classification of Facebook conversation data in previous sentiment analysis studies performed by the author indicated a lot of noise in the results. This worry was reinforced by the presence of a large number of actors with an international background in the data at hand, which introduced more variables in sentiment classification to adjust for.

Additionally, the challenge of achieving a good sentiment classifier without large financial commitments is beyond the scope of this thesis. For reasons depicted above, a search for a better visualization of conversation content than pure classification of Facebook user sentiment is prompted.

Language Analysis & Visualization A language analysis can be performed based on the conversation content which is included within the Facebook activity. For large datasets, language classification algorithms give a good estimate on the language which was used to create the social media artifact.

Challenges for language analysis are obviously some of the same that concern sentiment analysis. A problem is the heavy use of slang words and incomplete expressions within social media conversations which force practitioners into using tailor-made classification algorithms for each social media type, e.g. Facebook, Twitter and Instagram.

Apart from this, the challenges for language analysis are by far not as extreme as those for sentiment analysis. Furthermore, performant dictionary-based language detection algorithms are publicly available, thereby reducing the costs of performing a language analysis of the Facebook activity corpus.

Hence, the visualization of language analysis results should be part of the dashboard. This allows a grouping of social activity by language. Thereby, the language visualization may be materialized by using a traditional bar chart.

Word Frequency Analysis and Visualization One further analysis that can be performed based on conversation data from Facebook, is statistical word frequency analysis. A visualization of word frequencies during a previously selected timeframe is able to showcase the dominating topics of the conversation for a single Facebook wall or a set of Facebook walls.

Predominant visualizations of word frequency analysis results used in many visual analytics tools present bar charts and so-called word or tag clouds.

According to research performed by Hearst and Rosner (2008, p.8), alphabetical tag clouds "are to be applied to data representing human behavior, whether that of an individual or group". This visualization tools gives a fast overview over social activity in an appealing way; but when expressing transformation over time of non-social data tag clouds are not optimal from a data visualization perspective.

Furthermore, visual analytics tools used in (forensic) investigations, such as the *Jigsaw* tool evaluated by Kang et al. (2011) also contain alphabetical word frequency clouds for visualizing document and conversation content for human analysts. In this context, alphabetical word clouds were evaluated with very positive results.

Given these results, the decision was made to include word frequency analysis of conversation content as a visualization into the dashboard.

In this section, different types of visualizations for the Visual Analytics dashboard have been presented for the data at hand. Furthermore, several components have been analyzed from a design and analytics perspective. This provides important guidance for the implementation of the Big Data Visual Analytics tool.

4.4. Realization of the Visual Analytics Dashboard

At this point, I have proposed and outlined fundamental components of the Visual Analytics dashboard needed for an analysis of social media in combination with event timelines. The choice of these components is based on the assessment of the nature of the underlying data as presented in chapter 3.

In this section, I will analyze how a visualization dashboard as described previously can be implemented, and which commercial tools or frameworks are best to be used.

4.4.1. Discussion of Commercial Visual Analytics Tools

There are various commercially available tools and platforms which advertise the creation of Visual Analytics dashboards. Most of these tools are labeled as business intelligence solutions and marketed in the B2B space.

Subsequently, a number of commercial Visual Analytics tools were evaluated based on the requirements regarding input data sets and dashboard design which were established during previous chapters.

SAS Visual Analytics Tool The SAS tool is a partly web-based tool for visual big data analytics; but it is only commercially available and contains no specialized visualizations for social media data (Abousalh-Neto and Kazgan; 2012).

One of the more striking problems of the SAS tool is that it does not allow for alphabetically sorted word clouds which are needed for visualization of the results of the word frequency analysis of Facebook conversations. Additionally, the tool is not very extendable in terms of the application of further methods of interactive visualization.

Tableau The Tableau data visualization tool is very powerful and easy to use. During evaluation of tableau, a proof-of-concept implementation of the Visual Analytics dashboard in accordance to previously mentioned requirements was performed. The prototyping resulted with a negative outcome.

First, several visualization options posed problems because of their slowness when working with the large Facebook activity dataset. On top of that, the interactive components were hard to extend beyond the interfaces which they included out of the box. Furthermore, when using Tableau, the multi-device display of the Visual Analytics dashboard posed a problem. This is due to the automatically generated dashboard website, which could be accessed using a normal web browser, lacking many relevant features that were available in the stand-alone desktop version.

Downsides of Commercial Tools The previous discussion of commercial tools shows the existence of many shortcomings with regard to the implementation of a Visual Analytics dashboard for social media data. A rather large problem is the reliance of commercial tools on proprietary components which are hard to customize. This results in the need to circumnavigate problems that spontaneously arise from these components. This can happen any time during implementation or evaluation of the dashboard, and thereby poses a risk to the overall thesis project.

Research performed by Merkle et al. (2013) concludes that both tools are not suited for studies of dynamic and unstructured data. The author further underlines potential and proven problems that arise with the social media activity dataset used in this thesis.

On top of that, the *VizDeck* research project (Key et al.; 2012) also decided against using commercially available tools for implementation of the dashboard, as both Tableau and Google failed their preparatory evaluations.

4.4.2. Development of a Self-made IT Artifact

The discussion result of the previous section leads to the creation of a self-made IT artifact through which the Visual Analytics dashboard is implemented as a viable alternative to commercial tools. Hence, the choice to implement the Visual Analytics dashboard based on state-of-the-art open-source frameworks and programming libraries comes naturally and with plenty of positive effects.

A wide range of available tools and libraries is actively maintained by the open source community. Most of this software is based on best practices in handling of Big Data and in many cases an industry leading piece of engineering. Therefore, an open-source based approach presents a very productive environment for the implementation of a Big Data Visual Analytics dashboard, which - under these circumstances - can potentially be realized by a single programmer in scope of a thesis project.

Additionally, the implementation of an self-made piece of software for the dashboard and a production roll-out allow for higher levels of fine-tuning for each component. This in turn increases overall performance and ease of use which are set as major design objectives. A concentrated engineering effort can achieve a dashboard which is reliable when working with large social media and event timeline datasets.

Finally, the highly specialized requirement profile of a Visual Analytics dashboard as outlined in previous chapters is hard to satisfy without having full access to the underlying code base of the IT artifact. This depicts the most compelling reason for implementing the dashboard as part of this thesis.

4.5. Architecture

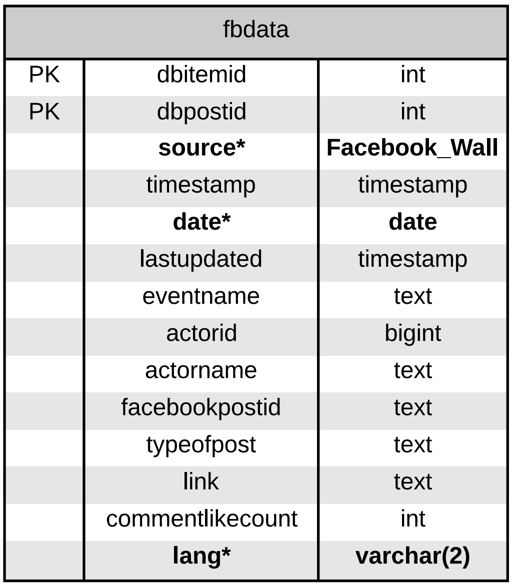
After thorough discussion regarding the best approach for realization of the Visual Analytics dashboard, a self-made approach is chosen in which an IT artifact will be developed as part of this thesis. In this section, information concerning the database and software architecture are presented in order to facilitate further understanding about the internals of the IT artifact.

4.5.1. Database Architecture

Within the future Visual Analytics dashboard, data needs to be effectively stored and queried. For this purpose, the two datasets described in section 3.1 are stored in a PostgreSQL database.

The Bangladesh event timeline, which has been described in section 3.2, is stored in a database table called bangladesh_events without any further modifications to the dataset.

Social media activity data from Facebook is stored in a database table, which is called fbdata and created with several optimizations in mind. Prior to storing the dataset, some modifications are performed in order to increase database performance for subsequent analysis tasks.



fbdata		
PK	dbitemid	int
PK	dbpostid	int
	source*	**Facebook_Wall**
	timestamp	timestamp
	date*	**date**
	lastupdated	timestamp
	eventname	text
	actorid	bigint
	actorname	text
	facebookpostid	text
	typeofpost	text
	link	text
	commentlikecount	int
	lang*	**varchar(2)**

Figure 4.1.: Table structure of SODATO-supplied Facebook data sets

In Figure 4.1, the layout of the fbdata database table is illustrated. It highlights that the social media activity data set received from *SODATO* needs to be modified in order to adapt it to the requirements of a Visual Analytics tool.

For this purpose, three new attributes are added to the fbdata table:

- The field **source** is added in order to distinguish between the origins of multiple pieces of social media activity. In particular, it depicts the company to which the respective Facebook wall belongs from which the activity is sourced.

- The field **lang** is the result of a preprocessing step performed as part of the *Big Data Pipeline* outlined in section 3.3 which employs a language classification algorithm on Facebook posts and comments in the social data. The algorithm returns a two-character length language identifier or *null* if language classification fails.
- The field **date** is derived from the timestamp information in the *timestamp* field. It is created in order to speed up database queries that heavily rely on date comparison.

4.5.2. Software Architecture

Prior to the implementation of a Big Data Visual Analytics dashboard, a viable software architecture needs to be decided upon. The dashboard relies on network communication between its server- and client-side components in order to transfer the data required for the visualizations.

Server-side Dashboard Component

The server-side component of the dashboard depicts a HTTP-based API backend, which also serves the client-side part of the dashboard and several media assets. The specific architecture of the server-side dashboard is of no further interest in this thesis as it basically consists of a standard web server implemented *Node.js* with additional connectivity to the *PostgreSQL* database.

Client-side Dashboard Component

The client-side component of the dashboard depicts a Javascript-based single-page web application, which enables the user to interactively use the Visual Analytics dashboard.

For means of implementation of the web application, decisions on abstraction levels and class inheritance of the dashboard's visualizations and views have resulted in the following software architecture illustrated in Figure 4.2.

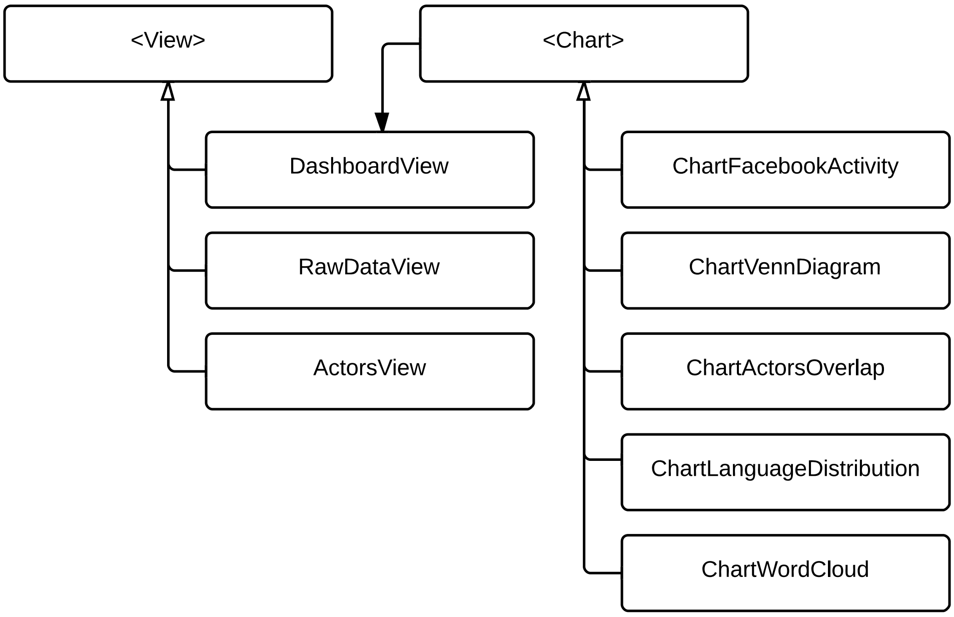


Figure 4.2.: Software Architecture

Views within the Dashboard Within the single-page web application, three different views are abstracted to a `View` metaclass which offers stateful navigation capabilities.

The implementations of the `View` metaclass are:

- **DashboardView**: This is the main view of the web application which contains the Visual Analytics dashboard. It is initially shown to the user.
- **RawdataView**: This view presents a detailed search interface for the Facebook activity data. Many visualizations in `DashboardView` refer to `RawdataView` in order to provide the user with further information.
- **ActorsView**: This view presents a dedicated interface for analysis tasks related to Actor Mobility as depicted in section 4.3.2. The visualizations of actor mobility in `DashboardView` refer to `ActorsView` in order to provide the user with further details when requested. Furthermore, `ActorsView` presents a handy set of tools for analysis of actor mobility and cross-postings between different time frames and facebook walls.

Visualizations within the Dashboard The Visual Analytics dashboard depicted by `DashboardView` includes five different visualizations.

Visualizations within `DashboardView` are abstracted by the `Chart` metaclass, which provides several convenience functions and stores visualization options as well as state. The most important customization option, which is available through the `Chart` metaclass, is a user-defined selection of the Facebook wall(s). Through this selection the user may explicitly choose the textile companies for which to display the visualization, thereby being able to apply a custom filter to each visualization.

All visualizations are organized as modules and may be independently customized by the user. The five visualization classes are:

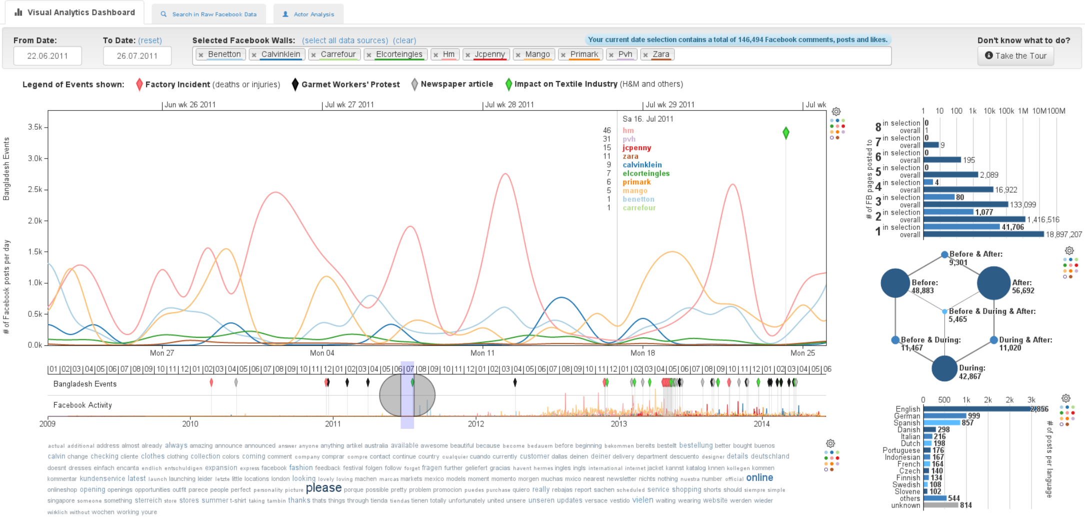
- **ChartFacebookActivity**: This is the main Facebook activity visualization as described in section 4.3.1.
- **ChartVennDiagram**: This is the first visualization of actor mobility as described in section 4.3.2. The visualization is based on a proportional Venn diagram which displays the overlap between different time periods.
- **ChartActorsOverlap**: This is the second visualization of actor mobility as described in section 4.3.2. The visualization is based on a bar chart which displays the overlap of social media activity by actors between different Facebook walls.
- **ChartLanguageDistribution**: This is the first visualization of conversation content as described in section 4.3.3. The visualization is based on a bar chart which displays the frequency of languages used in conversation.
- **ChartWordCloud**: This is the second visualization of conversation content as described in section 4.3.3. The visualization is based on an alphabetic word cloud which displays the results of a frequency analysis of word used in Facebook conversations.

To summarize this chapter, the design goals of the Visual Analytics dashboard and fundamental design principles have been introduced. Then, the major dashboard components and the underlying models have been presented.

After thorough discussion of the alternatives, the decision has been made to implement the Visual Analytics dashboard as a web application based on open-source frameworks. Furthermore, the server- and client-side software and database architecture of the dashboard have been illustrated. This means, we can now move on to the implementation of the Visual Analytics dashboard.

5. Development of the Visual Analytics Dashboard

This chapter examines the realization of the Visual Analytics dashboard, starting with the development process. Additionally, implementation details of the dashboard will be illustrated. Furthermore, the resulting IT artifact is presented.



From Date: 22.06.2011

To Date: (reset) 26.07.2011

Selected Facebook Walls: (select all data sources) (clear) Benetton, Calvinklein, Carrefour, Elcorteingles, Hm, Jcpenny, Mango, Primark, Pvh, Zara

Your current date selection contains a total of 146,494 Facebook comments, posts and likes.

Legend of Events shown: Factory Incident (deaths or injuries) Garmet Workers' Protest Newspaper article Impact on Textile Industry (H&M and others)

Figure 5.1.: Image of the final Big Data Visual Analytics Dashboard

Figure 5.1 displays an image of the final Big Data Visual Analytics dashboard which has been implemented during this thesis. The development of the respective IT Artifact will be described in this chapter.

5.1. Software Engineering Process

5.1.1. Requirements Engineering

Initially, requirements engineering has been performed based on the two data sources available for implementation of the dashboard. As previously noted, the Facebook activity dataset was generated using the SODATO tool. In terms of storage and visualization of basic social media activity, a precedent was set by the SODATO user interface as it already presented some visualization for the collected social data. This precedent was analyzed and improved upon where necessary.

Subsequently, a close feedback loop with a focus group of potential dashboard users was established, and weekly meetings and feature checks were performed. Besides others, the focus group included several academic peers at CBS's Computational Social Sciences Labroratory (CSSL).

After several meetings with the focus group of potential users, we agreed upon a basic set of features for the Visual Analytics dashboard, and the development of a proof-of-concept dashboard began. The prototype was implemented using the *Tableau* visual analytics software, but failed due to the shortcomings described in section 4.4.1.

Resulting from this failed prototype, the scope was increased into the creation of a custom-developed Visual Analytics dashboard for analyzing the Bangladesh event timeline and social media activity at hand.

The custom-developed Visual Analytics dashboard incorporates the following functional requirements, from which some have also been included within the design goals and objectives introduced in section 4.1:

- **Data**: Visualize the 89 Million Facebook events together with the 96 events of the Bangladesh event timeline in the time period from 2009 to 2014.
- **Visualizations**: Implement a customizable Visual Analytics dashboard with visualizations of social media activity, actor mobility and conversation contents. Create a detail on demand functionality.
- **User Goals**: The user can work with and customize the Visual Analytics dashboard. The user can search in raw Facebook activity data based on

all available criteria. The user can click on events in the Bangladesh event timeline and be shown details about the event.

- **Performance**: Dashboard components, including the backend, should be as fast as possible.
- **Accessibility**: Facilitate access from the following devices Tablet, Laptop and 4K conference room screen under the operating systems Windows 8, MacOS and Linux.
- **Responsiveness**: The dashboard should respond to changes in screen size and always aim at using screen estate as effectively as possible.

The following non-functional requirements are met by the implemented Visual Analytics dashboard:

- **User Interface**: The UI should be clean, usable and visually appealing.
- **User Experience**: The UX should be as pleasant as possible. The UI should give plenty of opportunities for customizations by the user.
- **Failure States**: The dashboard should fail gracefully in case of error.

5.1.2. Development Process

This section characterizes the general development process during the implementation of the Visual Analytics dashboard. Basically, the development process followed agile development principles and focused on frequent iterations and feedback loops with focus users.

Supportingly, a thorough testing of the client-side dashboard on different devices and screen sizes was performed ever since the beginning of development efforts. Additionally, testing of client- and server-side dashboard components was performed with respect to different constraints in terms of available memory, processor performance (single-core vs. multi-threading) and database backends.

These extensive tests facilitated the early detection of potential problems regarding architectural abstractions and performance bottlenecks, and resulted in a more testable overall software system.

5.1.3. Milestones

The following milestones were used during development of the Visual Analytics dashboard and have been met successfully:

1. **Finish the acquisition and preparation of both data sets** which are fundamental for the dashboard.
2. **Successfully import both data sets** into the database.
3. **Initial implementation of a baseline Visual Analytics dashboard** with only one visualization of Facebook activity.
4. **Achieve feature completeness** with regard to the functional requirements. These requirements resulted from the iterative requirements engineering process.
5. **Feature Lock** after implementing the non-functional requirements.
6. **Perform extensive load testing and performance optimization** for both server- and client-side components of the dashboard.
7. **Finish final version of the Visual Analytics dashboard** and deploy it for productive use in user evaluation.

5.2. Implementation

Implementation of the IT artifact is performed as previously outlined in a open-source environment. This allows for rapid prototyping during early development phases and scaling of software and databases during live deployments.

Due to the fact that the dashboard is required to work with multiple target devices, a browser-based implementation is chosen. Hence, the dashboard may be displayed on any device using a modern web browser and thereby achieving the accessibility requirement. Therefore, the dashboard is available on many different devices and form factors using common web application technology, such as Javascript, HTML5 and SVG.

The dashboard consists of a client-side and a server-side part. Both need to be implemented with focus on client- and server-side performance in mind. This is depicted by HCI-style ease of use and traditional scaling.

5.2.1. Programming Language

The server-side web application is based on the *Node.js* programming language created by Dahl (2012), which provides server-side Javascript. In contrast, the server-side application is mainly based on the *Express.js* framework that provides a basic HTTP server stack (Ihrig; 2013). HTML templates are created using the Jade templating language (Lindesay; 2012).

All client-side components of the Big Data Visual Analytics dashboard are implemented in Javascript that runs within the end users' web browser. The dashboard is reinforced by several programming frameworks such as *Backbone.js* which is used for view management. Furthermore, several convenience frameworks such as *underscore.js* (utility functions) and *moment.js* (date handling) are used.

All user-facing visualizations shown within the client-side Visual Analytics dashboard are implemented using dynamic SVG graphics which are built upon the powerful *D3.js* visualization framework.

5.2.2. Visualization Framework

The technology choice for realizing the dashboard visualizations is the *D3.js* Javascript-based visualization framework which uses dynamic SVG images for data visualization (Bostock; 2012).

D3.js constitutes a lightweight and very extendable Javascript visualization framework. It uses cutting-edge web technology in order to facilitate the creation of complex visualizations. Additionally, *D3.js* is actively under development by its original author, Mike Bostock. Bostock provides plenty of documentation which showcases many different visualization types and offers comprehensive best practices for *D3.js*-based visualization development.

The flexibility provided by *D3.js* enables the creation of new kinds of interactive visualizations which are able to run on any device with decent processing resources. This includes Windows, MacOS and Linux based systems with screen sizes up to 4K devices.

5.2.3. Dependencies

In the server-side component of the Visual Analytics dashboard, dependencies are specified within a **package.json** file. This displays best practice in *Node.js* web application development and allows easier integration with the module-centric Node.js package management system (NPM) created by (Schlueter; n.d.).

Listing 5.1: Dependencies of the server-side Visual Analytics Dashboard

```
{
"express": "^4.1.2",
"express-session": "",
"jade": "^1.3.1",
"pg": "",
"underscore": "^1.6.0",
"moment": "^2.6.0"
}
```

Server-side dependencies of the dashboard are showcased in Listing 5.1.

The dependencies are limited to a basic web framework (*express*), HTML templating (*jade*) libraries, and a PostgreSQL database driver provided by the *pg* module. The *underscore* and *moment* modules provide basic convenience functions for general-purpose tasks and date handling.

Listing 5.2: Dependencies of the client-side Visual Analytics Dashboard

```
/* Twitter Bootstrap CSS Framework */
./bootstrap.min.js
./bootstrap.min.css

/* Backbone.js MVC Framework */
./backbone.min.js
./backbone.queryparams.js

/* jQuery datatables */
./jquery.dataTables.min.js
./dataTables.bootstrap.js

/* utilities */
./jquery.min.js
./moment.js
```

```
./parallel.js
./jquery.lazyload.js
./underscore-min.js
```

Dependencies of the client-side web application are highlighted in Listing 5.2.

For creation of the web application which serves the dashboard, a Model-View-Controller approach is chosen and implemented using the *Backbone.js* framework. The software architecture is depicted in more detail in section 4.5.2

Furthermore, the *Twitter Bootstrap CSS Framework* is utilized to achieve a visually pleasant user interface which presents equally styled UI components to users on all devices (Otto and Thornton; 2010). In order to handle dynamic loading of large amounts of data within the `RawDataView` interface, the *jQuery Datatables* plugin is employed, which enables dynamic creation of HTML tables.

Listing 5.3: Dependencies of the client-side Visualizations as part of the Visual Analytics dashboard

```
/* d3.js */
./d3.v3.js

/* d3.js plugins */
./d3.hexbin.v0.js
./d3.layout.cloud.js
./d3.legend.js
./d3.tooltip.js

/* pretty HTML <select> items */
./select2.js

/* pretty date picker */
./bootstrap.datepicker.js

/* pretty loading bars */
./jquery.loadingbar.js

/* user tour */
./guidely.js
```

Additional dependencies of the client-side visualizations as part of the Visual Analytics dashboard are highlighted in Listing 5.3.

It becomes apparent that the visualization framework *D3.js* by Bostock (2012) plays an important role in the realization of visualizations within the dashboard. Further dependencies such as *select2.js*, *boostrap datepicker* and *guidely.js* provide methods of user interface improvement which are applied in order to reach a good end user experience when using the dashboard.

5.2.4. Performance

Naturally, performance is vital for interactive Big Data analytics tools such as the present dashboard. The performance requirement is even more important, when implemented in form of a web application which lacks native access to many resources of the underlying hardware. Based on this requirement, iterative steps of improving overall dashboard performance are undertaken. These steps include data preprocessing, aggressive caching of query results, and canceling of long-running processes when the user changes their navigational decision.

Furthermore, several sorting- and aggregation-related algorithms have been offloaded to the client-side components in order to gain responsiveness. This *offloading* does not harm usability of the dashboard as the client-side can process the received information in parallel by leveraging *HTML5 WebWorkers*.

Overall, the measures depicted above present a significant improvement in performance of the Visual Analytics dashboard.

5.2.5. Provisioning

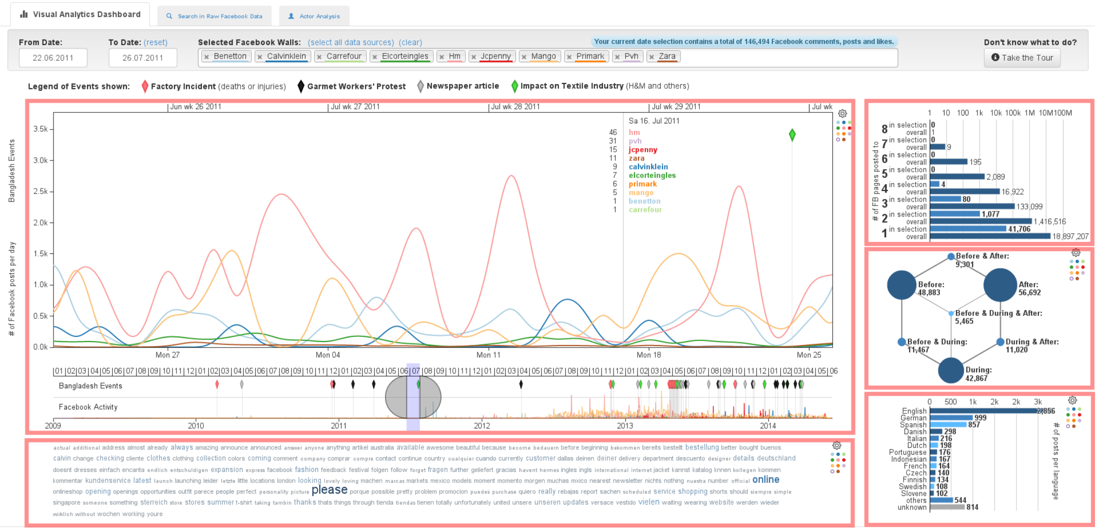
Provisioning of the Big Data Visual Analytics dashboard presented in this thesis can be performed in several steps.

First, the two datasets, the Facebook activity date and the Bangladesh event timeline, need to be imported into a database instance, preferably *PostgreSQL*. Second, several visualization-specific datasets are derived from the main dataset. This is a measure of performance improvement detailed in the upcoming section 5.5.2.

Finally, the server-side component of the Big Data Visual Analytics dashboard is launched and linked with the *PostgreSQL* database. The user is then able to connect to the dashboard by using a state-of-the-art web browser.

5.3. IT Artifact: Visual Analytics Dashboard

The IT Artifact representing the Visual Analytics Dashboard for Big Social Data is implemented as described in the previous sections. It is based on the thoroughly explained design decisions performed during the planning phase of this thesis.



From Date: 22.06.2011

To Date: (reset) 26.07.2011

Selected Facebook Walls: (select all data sources) (clear) Benetton, Calvinklein, Carrefour, Elcorteingles, Hm, Jcpenny, Mango, Primark, Pvh, Zara

Your current date selection contains a total of 146,494 Facebook comments, posts and likes.

Legend of Events shown: Factory Incident (deaths or injuries) Garmet Workers' Protest Newspaper article Impact on Textile Industry (H&M and others)

Figure 5.2.: Implementation of the Visual Analytics Dashboard with red rectangles depicting the areas of the five Visualizations

Figure 5.2 presents a high-level overview about the final composition of the Main Dashboard. The red rectangles are annotations that describe the five major visualizations on the Dashboard. These visualizations are based on the components introduced in section 4.5.2.

The **Facebook Activity** visualization displays the social media activity on Facebook over the whole time period. It consists of a large *main chart* and a smaller *mini chart* underneath. Both charts use a line plot to display activity. The *mini chart* can be used as a brush to change the time period of the data shown in the *main chart*.

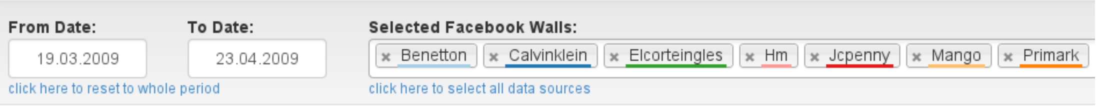
The **Actor Overlap between Facebook walls** visualization at the top right of the dashboard displays the number of different Facebook walls on which Actors have posted. For this visualization, a bar chart is used. The chart depicts the number of Actors based on the the number of Facebook walls they have posted to.

The **Actor Overlap between time periods** visualization at the center right of the dashboard displays the number of Actors within each time period and their respective overlaps. For this visualization, an exploded Venn diagram is used which is aligned hexagonally.

The **Language Distribution** visualization at the bottom right of the dashboard displays the number of social media Artifacts based on their language. For this visualization, a bar chart is used. It presents each language and the respective number of social media activities during the selected timeframe.

The **Word Cloud** visualization located right beneath the Facebook Activity chart displays the results of the word frequency analysis based on conversation Artifacts in the available social data. The font size of each word is determined by its overall frequency within all conversations that happened during the selected time period.

Furthermore, the Visual Analytics dashboard contains additional user interface components.



From Date: 19.03.2009
To Date: 23.04.2009
click here to reset to whole period

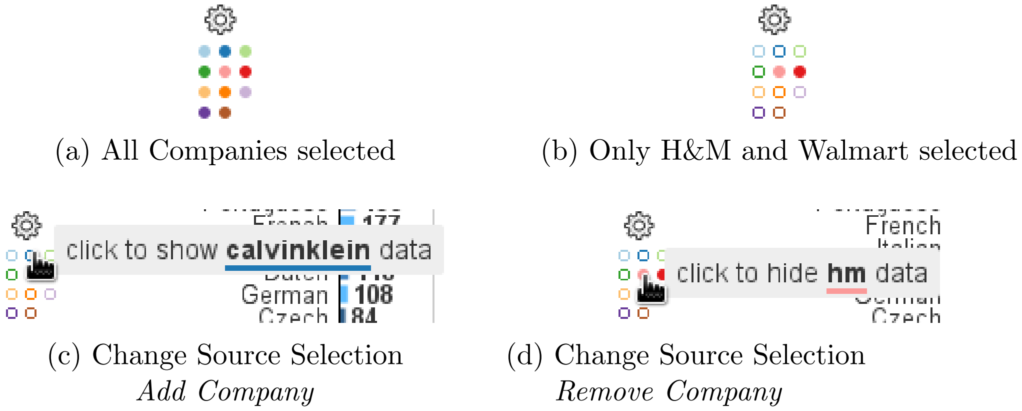
Selected Facebook Walls: Benetton, Calvinklein, Elcorteingles, Hm, Jcpenny, Mango, Primark
click here to select all data sources

Figure 5.3.: User-driven filtering interface in the Visual Analytics dashboard

Figure 5.3 displays available methods of filtering the Big Social Data displayed in the Visual Analytics dashboard. Filtering can be performed by the user based on timeframe and source.

The **user-driven filtering interface** contains two components. On the left hand side, the user may input *Start* and *End Date* of the timeframe to be visualized in the dashboard. Mouse or touch interactions with the input fields will reveal a hidden date picker component based on *Boostrap Datepicker*. This date picker enables the user to either input dates using a keyboard or specifying the day, month and year using their mouse or even a touch screen.

Secondly, on the right hand side, the user may select the companies whose *Facebook walls* are shown in the Visual Analytics dashboard. User interaction with the available input field can be performed in various ways. The user can directly type Facebook walls into the field, which are then displayed in the visualization. An alternative method is that the user selects an item from a dropdown menu that appears when the input field is focused.



(a) All Companies selected

(b) Only H&M and Walmart selected

(c) Change Source Selection *Add Company*

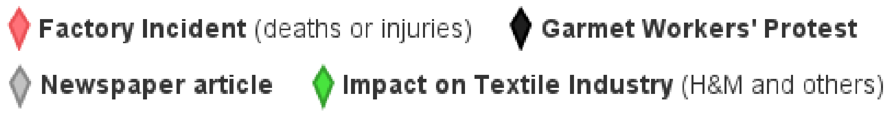
(d) Change Source Selection *Remove Company*

Figure 5.4.: Dot-based filtering interface for each Visualization

Mini Facebook Wall Selection In addition to the selection possibility of Facebook walls through the *user-driven filtering interface* described above, another method for selection of data sources exists. This additional interface is shown in Figure 5.4. It is rather unobtrusive and therefore easy to miss by a certain kind of users. As the illustration shows, interaction patterns for adding and removing sources for a visualizations are possible, as (c) and (d) show.

The dot-based filtering interface is available for each visualization. It is located in the top right corner of each chart, and depicts the current source selection state as shown in Figure 5.4 (a) and (b). Furthermore, a click on the gear icon at the top of the dot-based source selection shows a more verbose dialog for changing of the selected Facebook walls for a certain visualization.

Legend for Event Timeline Another crucial piece of information for dashboard users is shown in Figure 5.5. This legend is placed at the very top of the



Factory Incident (deaths or injuries) **Garmet Workers' Protest**

Newspaper article **Impact on Textile Industry** (H&M and others)

Figure 5.5.: Different types of Events as depicted in the Legend for the Bangladesh Event Timeline

dashboard between the user-driven filtering interface and the Facebook Activity visualization. It conveys information about different types of events which are part of the event timeline. In the case at hand, the event timeline is based on the Bangladesh factory disaster events, which means that the event types classified in section 3.2 are encoded in the legend.

To summarize, the Big Data Visual Analysis dashboard empowers users to use it in different ways. The dashboard adheres to the user's preferred interaction method without making any assumptions. This means tablet users may also type in their selection of the Facebook walls, or desktop users may use the datepicker to manually select a date.

5.3.1. Line Chart: Facebook Activity over Time

In the previous section, the Facebook Activity chart was briefly introduced. It consists of two charts, the *main chart* and the *mini chart*, which will both be explained in more detail.

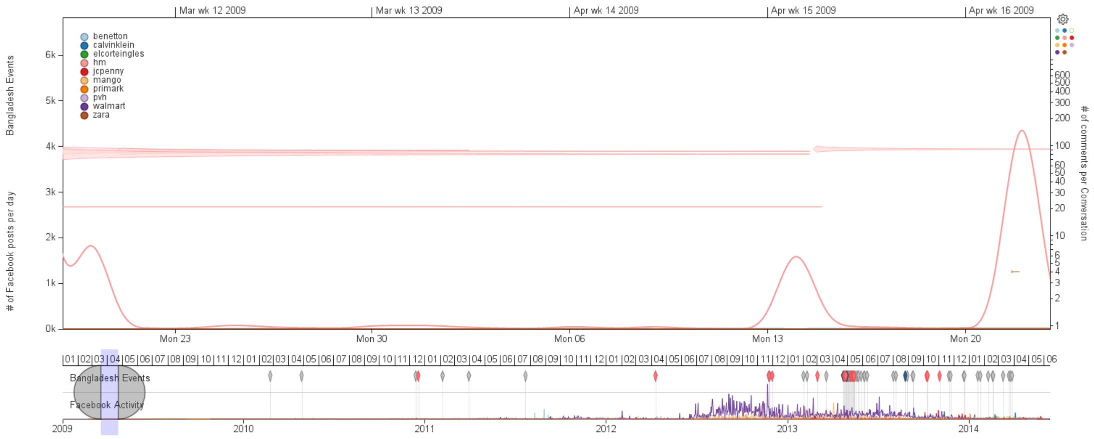


Figure 5.6.: Line Chart Visualization of Facebook Activity in the Dashboard

Main Chart In Figure 5.6, the main chart is shown as taking up the majority of space at the top of the Facebook Activity visualization. The main chart consists of a line chart displaying the Facebook activity on a per-day basis. Two axes, both at the top and the bottom, provide information about the currently selected date period. The line chart contains an optional legend at the top left. The chart's legend also explains the color coding of the companies' Facebook walls.

The left axis of the main chart states the number of social media activity for each day, and scales dynamically depending on the maximum daily social media activity within the selected timeframe.

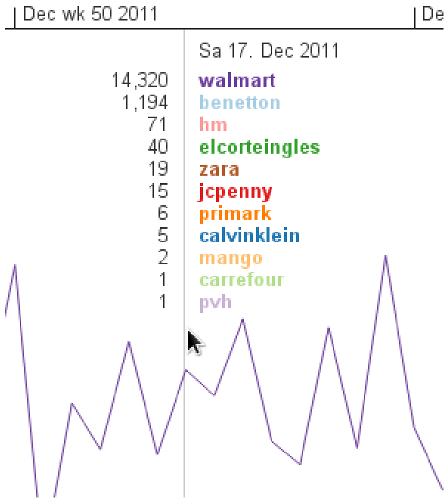


Figure 5.7.: Dynamic Inline Legend displayed in the Facebook Activity Visualization

Figure 5.7 presents another component of the user interface which is only shown when specific conditions are met. The dynamic inline legend is displayed when mouse interactions with the main chart are performed by the user. Its placement follows the user's mouse movements. The main information presented to the user through this legend is the date of the currently focused day in the visualization. Additionally, the number of social media activity per company on that very date.

Mini Chart The mini chart is placed at the bottom of the Facebook Activity visualization as shown in Figure 5.6. It allows the user to navigate the whole period of available data by using a specifically created brush handle. Using the brush handle, users can set the time period and detail level of the main chart.

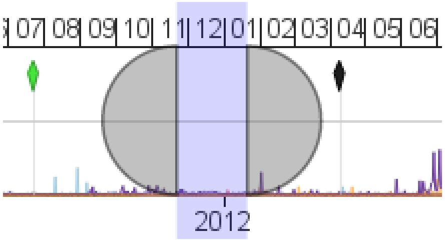


Figure 5.8.: Brush Handle for Panning and Zooming of Facebook Activity Data

The brush handle is demonstrated in Figure 5.8. It can also be modified by the user through use of mouse wheel scrolling and touch events.

Furthermore, the mini chart gives a complete overview over the whole timeline of Bangladesh Events by using color-coded event markers. For this purpose, small representations of the event markers presented in Figure 5.5 are shown in the mini chart and placed accordingly over the whole time period. The user can interact with the event markers presented in the mini chart, thereby automatically zooming in to the day of the event using the main chart.

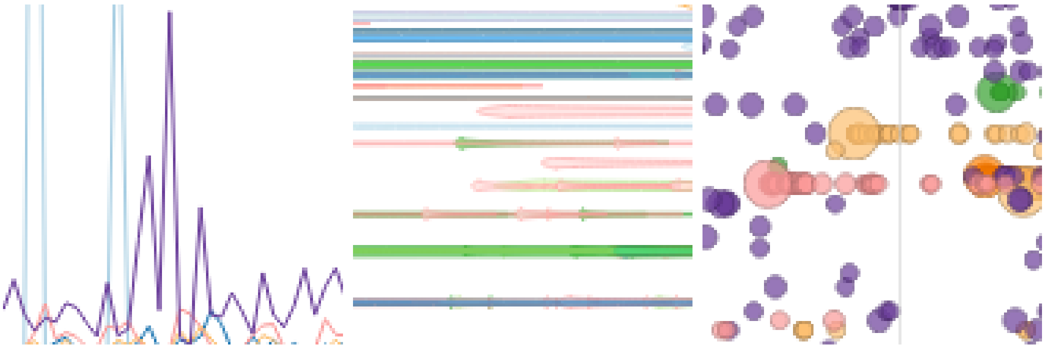
Zoom-driven Detail Levels In accordance with the *Detail on Demand* principle in visual analytics, which was already introduced in the course of this thesis, the main chart may be zoomed in by the user in order to increase the level of detail shown.

If the level of detail set by the user is high enough to show additional visualizations of social media activity, a right axis will be shown. The right axis then displays the number of comments in each conversation. Furthermore, the line chart will have additional dot- or comment-style visualizations depicting single pieces or strings (conversations) of social media activity.

The general detail level of the main Facebook activity chart is determined by the user-defined zoom level and the number of total Facebook activity in the selected time period. Those two factors are chosen in order to prevent performance bottlenecks on the users' web browser when working with the dashboard.

Facebook activity in the main chart can be displayed in three different levels of detail - low, medium and high - which are illustrated in Figure 5.9.

The initially shown line chart of Facebook activity depicts the lowest level of detail.



(a) Line Chart *Low Detail*

(b) Conversation Comets *Medium Detail*

(c) Post/Comment Dots *High Detail*

Figure 5.9.: Example of the three Levels of Detail of the Main Facebook Activity Visualization: Low (a), Medium (b) and High (c)

The next higher level, medium detail, is displayed in 5.9 (b). It consists of **Conversation Comets** which visualize conversations based on their overall length, likes and the number of comments within the conversation.

The highest level of detail can be reached by zooming into the main chart as far as possible. It is depicted by so-called **Comment Dots** in which each social media Artifact is displayed by a single dot. The size and position of the dot depends on the likes it has received and the conversation it belongs to.

Both *Conversation Comets* and *Comment Dots* are interactive and may be clicked hovered or clicked by the user. When user interactions are performed, more details on the conversation or the social media Artifact will be shown.

As outlined above, the decision on the level of detail which is shown in the main chart depends on two factors. After each zooming or panning interaction with the Facebook Activity visualization, a calculation is performed in order to decide on the optimal level of detail in the visualization.

The level of detail of the main activity chart can therefore be defined as

$$detail\_level(D, N) = \begin{cases} SHOW\_DOTS, & \text{if } N \leq 5 * 10^3 \text{ or } D \leq 5 \\ SHOW\_COMETS, & \text{if } N \leq 3 * 10^4 \text{ or } D \leq 10 \\ SHOW\_LINES\_ONLY, & \text{otherwise} \end{cases}$$

where D is the total number of days selected, and N is the total number of Facebook activity in the current user-selected timeframe.

5.3.2. Bar Chart: Language Distribution

In section 4.3.3, we have already discussed the preferred visualization options for conversation content.

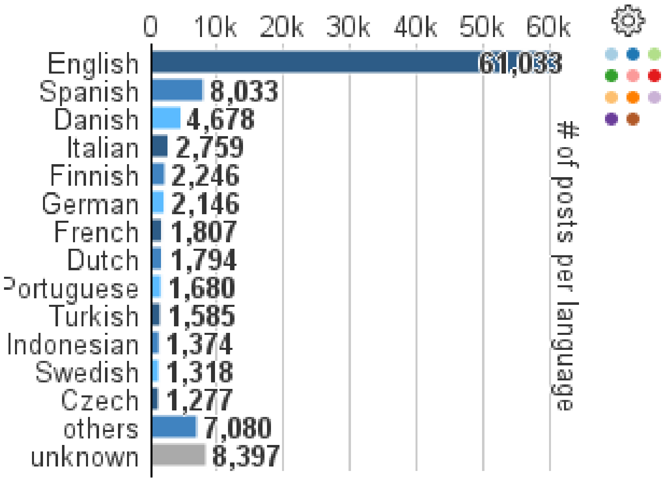


Figure 5.10.: Visualization of Language Distribution

Figure 5.10 showcases the visualization of the Language Frequency Analysis results. The automatically performed frequency analysis employs the user-defined timeframe to select all conversation artifacts from within that timeframe. Each comment and post in the conversation is then classified using a naive language classifier, and all occurrences of a specific language are recorded. In the end, a sorted list of tuples of `(language, number_of_posts)` is returned to the visualization.

The visualization uses data resulting from frequency analysis to create a bar chart which displays the most used languages within conversations during a specific, user-defined timeframe.

5.3.3. Word Cloud: Word Frequency Analysis

Another visualization of conversation content describes the alphabetical Word Cloud. It visualizes the most used words in an easily comprehensible manner and displays their respective frequency over the selected timeframe.

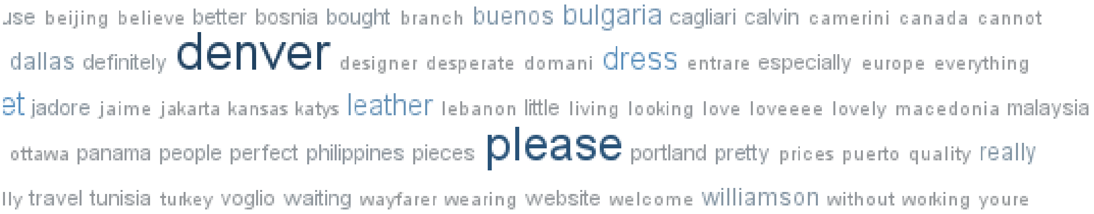


Figure 5.11.: Alphabetic Word Cloud Visualization of a Word Frequency Analysis

Figure 5.11 depicts the word cloud visualization. For a better visualization of actual conversation content, several adjustments have been made regarding to the inclusion of words into the word cloud. For one, website urls, emoticons and very short words have been removed. On top of that, all company names were hidden in order to not distort the visualization too much[1]. Lastly, several special characters have been removed from the word cloud. This was done in order to combat the problem of common misspellings creating visual clutter within the word cloud.

5.3.4. Bar Chart: Actor Overlap between Facebook Pages

In section 4.3.2, the concepts of Actor mobility analysis were introduced. In the Visual Analytics dashboard, these concepts are incorporated into a visualization of actor overlap between different Facebook walls. This visualization enables dashboard users to see how many social media Actors performed Actions on one or multiple Facebook walls during the selected timeframe.

Figure 5.12 showcases the resulting visualization, in which Actors are shown that have created Artifacts on one to eight of the eleven examined Facebook walls. The maximum number of Facebook wall cross postings by a single Actor is eight,

[1] In fact, “Walmart” depicts the most frequently used word over the whole time period, therefore it presents not much value as a signal for visual analytics. Hence, it was removed.

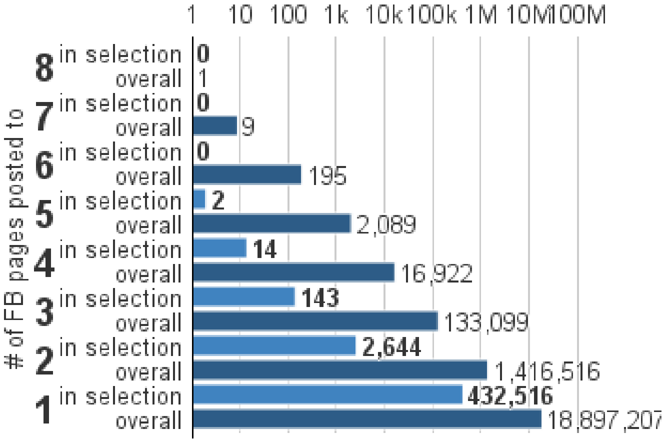


Figure 5.12.: Visualization of Actor Overlap between different Facebook Walls

which has been determined by analyzing the whole dataset. This means that even though there are a total of eleven Facebook walls included in the Visual Analytics tool, no single Actor has posted on nine, ten, or all eleven of the walls. The data for this visualization is calculated based on an analysis of all actors for each selected Facebook wall and a subsequent overlap calculation between the sets for each wall.

5.3.5. Venn Diagram: Actor Overlap between Time Periods

The dashboard contains a Venn Diagram visualizing actor overlap between different timeframes, which are *Before*, *After* and *During* the current user-selected timeframe.

In Figure 5.13, the rather unconventional Venn diagram is shown. Due to problems with the calculation of proportional Venn diagrams, a hexagonal split of the overlap areas of the Venn diagram has been performed. These specific problems only arise when different overlap areas express hugely varying numbers, under conditions such as with the Big Social Data at hand. Caused by this problem, the visual appeal of a conventional Venn diagram, which exhibits overlapping areas, is largely reduced for visual analytics.

A solution to this problem has been found by displaying each overlap region as a circle arranged in a hexagon. The links between circles signify the overlapping parts. This way, the analytical value of the visualization could be restored.

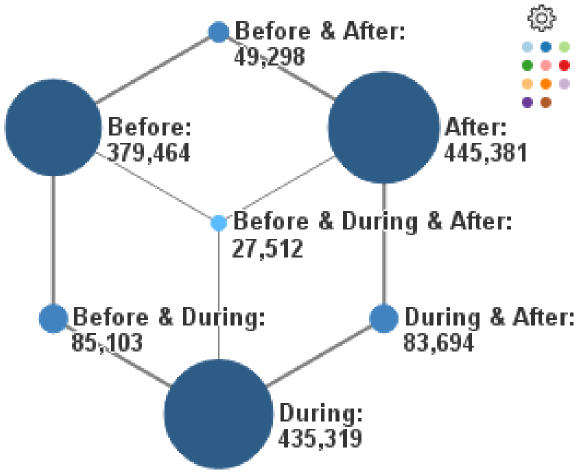


Figure 5.13.: Visualization of Actor Overlap between different time periods: Before, During and After

5.4. IT Artifact: Auxiliary Views

In addition to the Big Data Visual Analytics dashboard, several auxiliary views present opportunities for deeper analysis of the underlying datasets. Hence, when interacting with individual components of the dashboard, the user is in many cases redirected to one of the auxiliary views. These views are able to give much more detailed information than the visual interface of the dashboard could.

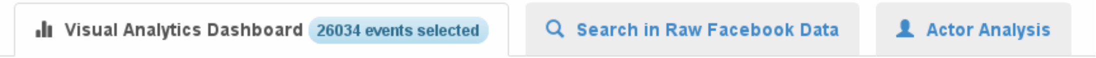


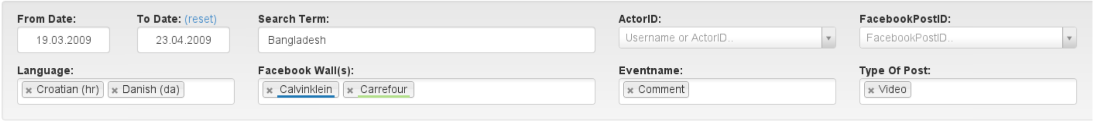
Figure 5.14.: Navigation Interface for switching between the Dashboard and Auxiliary Views

In Figure 5.14, the navigational controls of the dashboard are presented. Using these controls, the user can switch between the Visual Analytics dashboard and other views at any time.

When switching between views, the users' current state is captured and memorized. In case the user decides to return to the previous interface, the previously memorized state is restored. This enables the user to continue working on the interface in the same mental state as when he navigated away from it.

5.4.1. Raw Data View

By using the *Raw Data View*, the dashboard user can explore and filter the raw datasets using any attributes present in the database.



From Date: 19.03.2009

To Date: (reset) 23.04.2009

Search Term: Bangladesh

ActorID: Username or ActorID..

FacebookPostID: FacebookPostID..

Language: × Croatian (hr) × Danish (da)

Facebook Wall(s): × Calvinklein × Carrefour

Eventname: × Comment

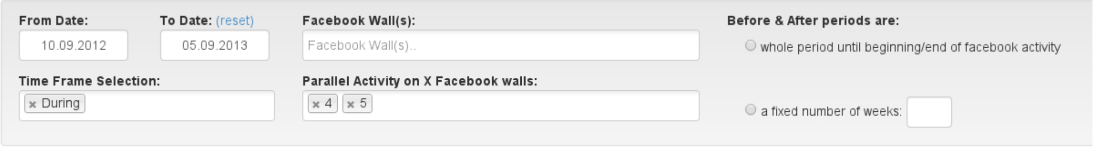
Type Of Post: × Video

Figure 5.15.: Data Filtering in the auxiliary Raw Data View

As depicted in Figure 5.15, several user interface improvements have been performed in order to allow for a better user experience. Hence, comparable input fields are offered to the user from which the latter may seamlessly choose their preferred methods of interaction.

5.4.2. Actor Analysis View

The *Actor Analysis View* presents detailed information on the underlying data of the actor analysis visualizations outlined in sections 5.3.4 and 5.3.5.



From Date: 10.09.2012

To Date: (reset) 05.09.2013

Facebook Wall(s): Facebook Wall(s)..

Before & After periods are:

○ whole period until beginning/end of facebook activity

○ a fixed number of weeks:

Time Frame Selection: × During

Parallel Activity on X Facebook walls: × 4 × 5

Figure 5.16.: Data Filtering in the auxiliary Actor Analysis View

Figure 5.16 emphasizes that the Actor Analysis view presents a comprehensive analysis interface, which is comparable to the one presented in the *Raw Data View*. Using the *Actors Analysis View*, extensive research can be performed on multi-dimensional actor overlap within the social activity dataset from Facebook. Complex issues can be investigated based on the number of Facebook walls to which an Actor has posted to, a set of Facebook walls to inspect, or a specification of overlapping time periods.

5.4.3. Detailed Facebook Activity View

The *Detailed Facebook Activity View* displays standalone information about social media artifacts from the Facebook data. The view is implemented as a modal dialog which can be shown using interaction with the Facebook Activity visualization.

5.4.4. Detailed Event Information View

The *Detailed Event Information View* provides details of events from the Bangladesh Event timeline. The view plays an important role in facilitating detailed access to the event timeline outlined in 3.2. It is implemented as a modal dialog which can be displayed by clicking on the event indicators shown in the main chart of the Facebook Activity Visualization.

5.5. IT Artifact: Server-side Components

The server-side artifact works with both datasets outlined in previous chapters and communicates the visualization-ready results to the client-side Visual Analytics dashboard. This distribution of workload aims at using the available resources in the most efficient way possible. Naturally, the server-side displays more optimization potential performance wise. This is caused by the lack of control over the client-side processing environment of the dashboard, not only on closed systems such as devices, but also for regular users of the Visual Analytics dashboard.

As depicted previously, the webserver is built upon the *Express.js* framework (Ihrig; 2013) and uses the *Jade* templating engine (Lindesay; 2012) for generation of HTML-based responses. The webserver provides the API and runs single-threaded.

5.5.1. Network Communication & API

The server-side API is HTTP-based and uses JSON for data transfer. Server- and client-side components are in constant communication during active use of the dashboard.

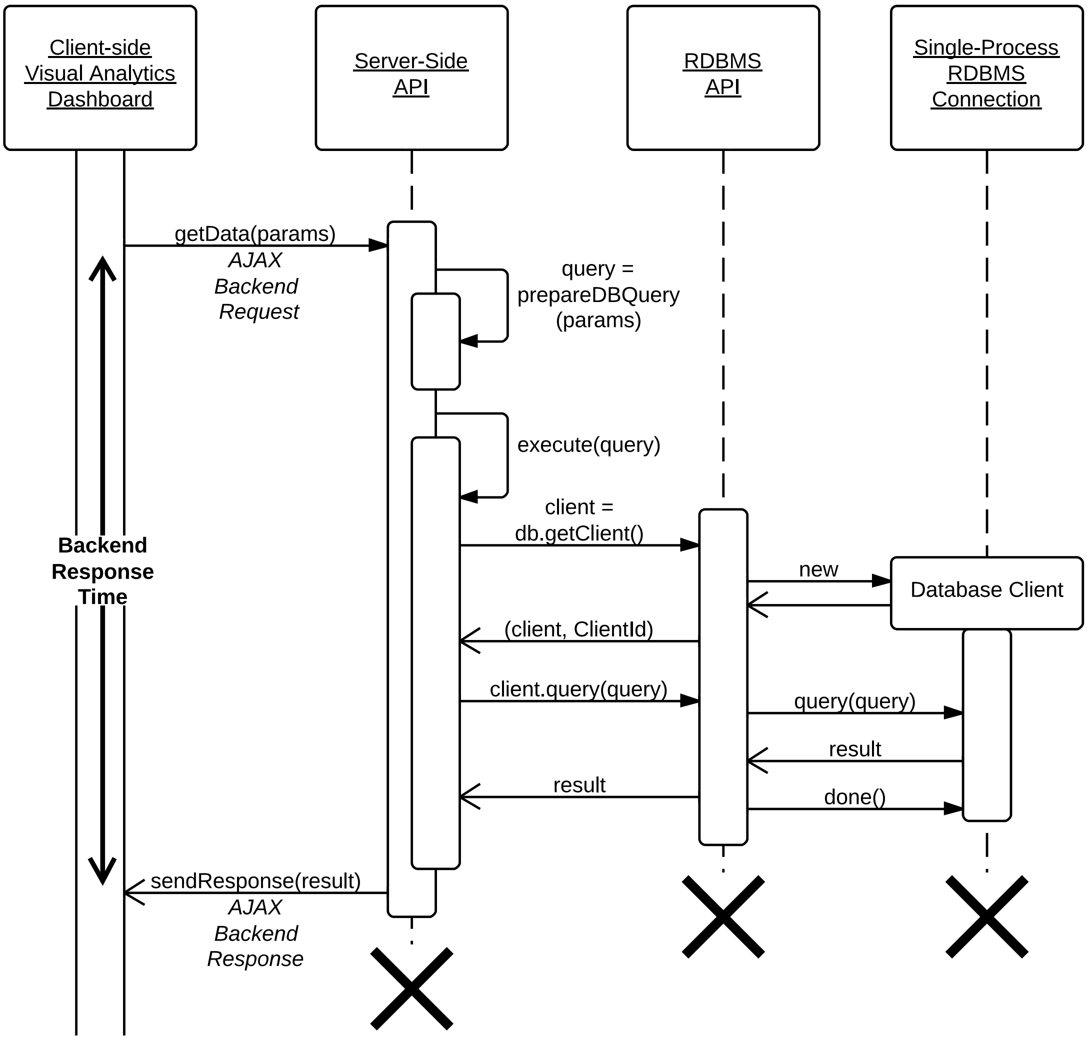


Figure 5.17.: Overview of Network Communication between Server- and Client-side Dashboard Components

Figure 5.17 illustrates the network communication which is performed between server- and client-side components of the Visual Analytics dashboard in an abstracted way.

The main actors in network communication are the *Client-side Visual Analytics Dashboard*, the *Server-side API*, the *Database Connection Layer* and the actual *Database*.

During normal use of the Visual Analytics dashboard, the client-side component will request data from the server-side component using the provided API. Based on the request, the server-side API prepares a database query for the requested data.

The server-side component also creates a database connection using the Database Connection Layer (or RDBMS API), which is then returned to the server-side component in form of a `Database Client` object instance.

Using the *Database Client*, the server-side component can finally request the actual *RDBMS* for the data it needs. The database query is then executed by the RDBMS and the result is returned to the server-side component, which can then in turn process the data.

After the server-side component finishes its additional data processing, the result is returned to the client-side component and displayed in the dashboard.

From thoroughly analyzing this process, it becomes apparent that *Backend Response Time* represents the most important networking metric of the Visual Analytics dashboard. Hence, in order to facilitate a smooth working with the dashboard, *Backend Response Time* needs to be minimized.

5.5.2. PostgresSQL Database

The previous section investigated challenges in network communication, this section will concentrate on basic means of improving the overall performance of the dashboard by enhancing the database configuration, data access patterns and layout for every single visualization.

Database Configuration The out-of-the-box configuration of the employed *PostgreSQL* database needs to be customized according to the opportunities presented by the underlying system.

Listing 5.4: Configuration Changes to the production PostgreSQL database

```
shared_buffers = 128MB  # previously: 2MB
temp_buffers = 24MB     # previously: 1MB
work_mem = 1000MB       # previously: 10MB
```

Therefore, several configuration parameters are changed in accordance with best practice documentation of PostgreSQL in order to use the 32 GB working memory of the production environment of the Visual Analytics dashboard. The changed configuration is depicted in Listing 5.4.

Database Performance Engineering Based on the initially presented database architecture of both datasets as described in section 4.5.1, further improvements seem necessary. Unfortunately, the rather small size of the Bangladesh event timeline dataset makes it not very relevant for further database optimization steps. Hence, in terms of database performance engineering, there is more room for significant improvements in the handling of the Facebook activity dataset, which contains nearly 90 million table rows.

Consequently, methods for improving the Facebook activity dataset, which is stored in the `fbdata` table, will be presented in this section.

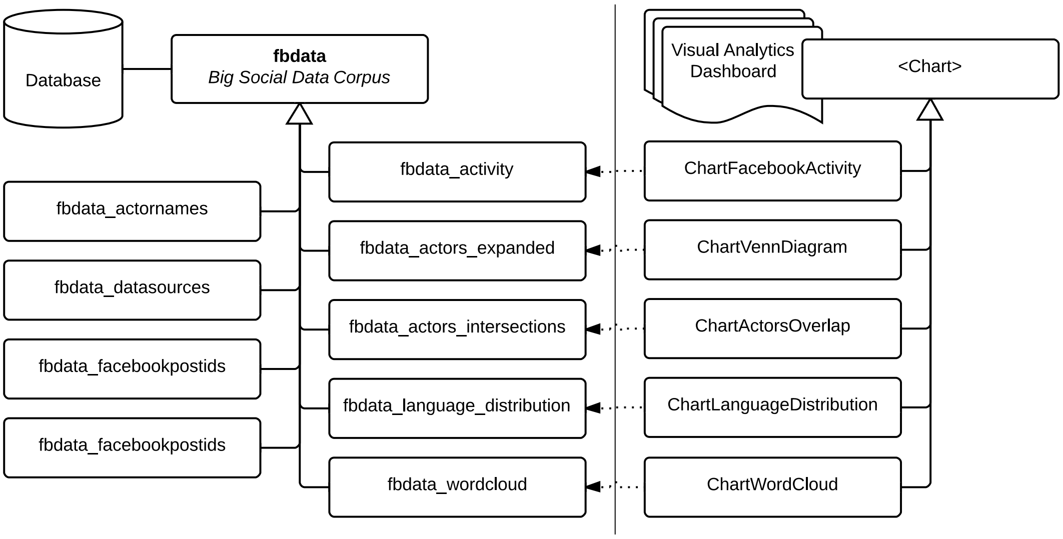


Figure 5.18.: Database Architecture after Performance Improvements through Derived Tables

Figure 5.18 provides an overview of all database optimizations that have been performed based on the original `fbdata` table. In this figure we can see the main Facebook activity data set has been derived into a total of nine different database tables, thereby leveraging an optimization technique based on dataset derivation.

The basic idea behind dataset derivation is that the original, very large dataset is split up into smaller, dedicated datasets. Figure 5.18 illustrates that the `fbdata` table is used to create one dedicated table for each of the five different visualizations of the Visual Analytics dashboard.

Thus, as part of an one-time effort, the following database tables are created in order to improve the performance of the visualizations of the Visual Analytics dashboard:

- Database table `fbdata_activity` is created in order to improve the line chart within the Facebook Activity visualization.
- Database table `fbdata_actors_expanded` is created in order to improve the actor overlap calculations as part of the Actor Overlap Venn diagram.
- Database table `fbdata_actors_intersections` is created in order to improve the actor overlap calculations of the Actor Overlap Bar chart.
- Database table `fbdata_language_distribution` is created in order to improve the Language Frequency Analysis visualization.
- Database table `fbdata_wordcloud` is created in order to improve the word frequency analysis for the Word Cloud visualization.

Additionally, several other database tables are created to improve one-variable lookups in the `fbdata` dataset, which are needed for several user interface improvements of the auxiliary views described in section 5.4.

The presented engineering feat improves overall performance of the visualization dramatically. However, this comes at the expense of additional disk space that is needed for storage of the newly generated derived database tables. The driving force behind these noticeable improvements is the basic fact that, when using the derived tables and not the full `fbdata` dataset, the RDBMS simply needs to process much less data for many database queries during dashboard runtime.

Logically, the `fbdata` table has still to be queried in many cases, such as when the end users requests a higher level of detail in the Facebook activity chart or when a raw data search is performed - so not all load can be relieved from the database.

6. Evaluation

At this point, a deep understanding of the design and implementation details of the Visual Analytics dashboard has been established. But, further evaluation steps are needed to critically assess the achievement of the goals and objectives stated beforehand.

The evaluation of a Visual Analytics Dashboard can be performed in various ways, such as software-, database- or user-focused testing. The following sections will elaborate on the tests performed with the Visual Analytics dashboard and their respective outcomes.

6.1. Software Testing

Software testing is commonly performed in various ways, and based on functional requirements or conventional unit testing. With these methods, test coverage statistics are often used to portray an assessment of the software at hand.

Furthermore, the interaction between the software artifact and the utilized database needs to be tested. This can best be done by performing a diverse set of benchmarks and load tests of common usage scenarios.

6.1.1. Functionality Testing

The functionality of the IT artifact presented in this thesis has been tested on an ongoing basis with a group of core users. In reliance on the previously defined requirements (see 5.1.1), a continuous feedback loop is established. The goal of this feedback loop is to receive a fully functional Visual Analytics dashboard in the end. Error or feature completion reports are communicated in form of direct user feedback over a multitude of channels. These communication channels could be either established within meetings, email or video conferences.

6.1.2. Conventional Unit Testing

Conventional unit testing was previously established as one of the most common software testing methods. In many cases, unit testing relies on well-defined interfaces, edge cases or malformed input.

The major problem with unit testing during the development of the Visual Analytics dashboard is that dynamic user interfaces are generally very hard to test. This is caused by the large number of different states a UI can represent and the numerous interaction methods that are available to the dashboard user.

Therefore, during development of the Visual Analytics tool, conventional unit testing was limited to the server-based software routines which handle the datasets. In this realm, the data handling scripts of SODATO-provided social activity from Facebook make heavy use of unit testing. This is done in order to provide a formal correctness of the resulting data, which is the foundation of the Visual Analytics Dashboard.

6.1.3. Software Benchmarking

During the development of the dashboard, software benchmarking presented an ongoing instrument in optimization of all software-centric processing routines.

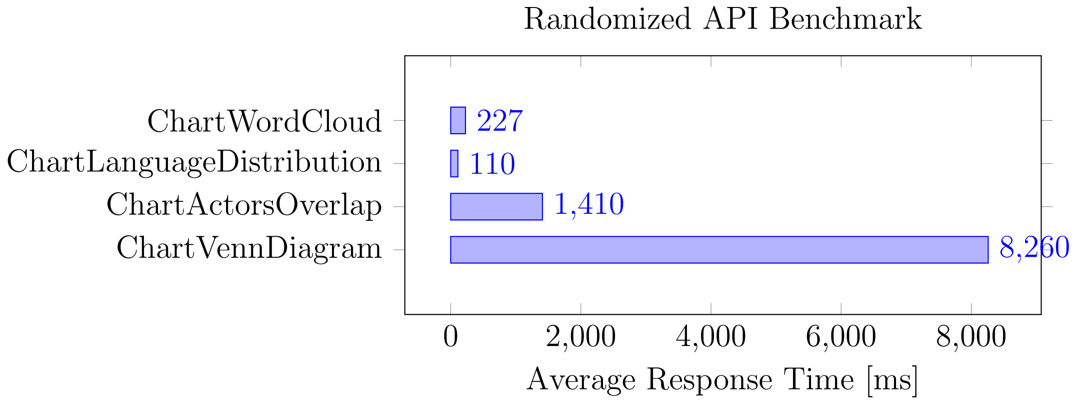


Figure 6.1.: Performance Benchmark of four API Endpoints of the Visual Analytics Dashboard

Benchmark Results Figure 6.1 displays the results of the randomized API benchmark tests. The results underline the varying complexity in calculating the data for the visualizations. According to the presented benchmark, visualizations of conversation content (*ChartLanguageDistribution* and *ChartWordCloud*) are much faster calculated and presented to the dashboard user than visualizations of actor mobility (ChartVennDiagram, ChartActorsOverlap).

This can be explained by the fact that visualizations of Actor Mobility need to take each single user into account, whereas the conversation-content visualizations have access to much better speed improvements by tables derived from the main `fbdata` dataset.

Due to the bad benchmark results of *ChartVennDiagram*, and a general discrepancy in performance, further optimizations are performed to the database as described in section 6.2.2.

Methodology The presented response time data has been collected and averaged over 100 subsequent requests to each of the four major API endpoints in the server-side backend[1]. The queries have been performed one after the other. The normally user-selected timeframes were randomized over the available time period and had a randomly selected length in the interval of $[5, 60]$ days. This was done in order to represent realistic user behavior. In all cases, the visualizations were calculated with respect to all data sources (Facebook walls).

6.2. Database Testing

Subsequent to achieving feature completeness in the development of the Visual Analytics dashboard, extensive database testing has been performed. Based on database testing, the accomplishment of non-functional requirements stated in section 5.1.1 is reviewed.

Thus, a verification of the non-functional performance requirement in regard to all database-dependent visualizations on the client-side dashboard is performed. This verification can be conducted by extensively testing the database and database-facing parts of the server-side component. If any performance bottlenecks become apparent, an optimization has to be carried out in order to reach an optimal outcome.

[1] No API Backend for `ChartFacebookActivity` exists, therefore it is not tested.

6.2.1. Query Optimization

When using a RDBMS such as PostgreSQL in Big Data analytics, many opportunities for increased performance can be realized through query optimization. The systematic optimization of slow database queries will be demonstrated on the visualization of Language Distribution.

Methodology All optimizations will be benchmarked against the initial query in order to assess their effectiveness. The benchmarking process follows a strict methodology, in which each query will be executed $n = 10$ times and query execution time is logged. Then, the average execution time is used to decide on the feasibility of the optimization at hand. If the average execution time is reduced, the optimization step will be applied to the query. The optimization process may be repeated until sufficient reduction of the average query execution time is reached.

Example: Optimization of Language Distribution Query

The language distribution query was identified as a rather slow query. Therefore, the original query becomes the initial test case, from which the optimization process is started.

Listing 6.1: Initial Query for Language Distribution

```sql
SELECT lang,
 COUNT(*) as count
FROM fbdata
WHERE
  eventname != 'LIKE' AND
  "date" BETWEEN '2009-01-01' AND '2014-06-12'
  AND source in ('carrefour', 'walmart')
GROUP BY lang ORDER BY count DESC;
```

The initial query as displayed in Listing 6.1 returns 24 rows after roughly 10 seconds. This does not meet the end user's performance expectations for a Visual Analytics dashboard. Therefore, the query needs to be optimized.

Based on the technique of deriving a new table from a larger one, we can derive a new database table fbdata_language_distribution from the fbdata dataset (see section 5.5.2 for further details).

Listing 6.2: First Optimization of Language Distribution Query

```sql
-- create a derived table to speed up language distribution
CREATE TABLE fbdata_language_distribution AS
  SELECT date, source, lang, count(*) as count
  FROM fbdata
  WHERE eventname != 'LIKE'
  GROUP BY date, source, lang
  ORDER BY date, source, lang ASC;

-- further queries only use the derived table
SELECT lang, sum(count) as count
FROM fbdata_language_distribution
WHERE
  "date" BETWEEN '2009-01-01' AND '2014-06-12'
  AND source in ('carrefour', 'walmart')
GROUP BY lang ORDER BY count DESC;
```

Listing 6.2 depicts the first optimization step which is performed based on the derivation of a new database table from the large dataset. When querying the derived table instead of using the initial query from Listing 6.1, we get a measurable performance improvement.

This performance improvement is based on the fact that the new query does not need to access the much larger *fbdata* table, but only uses a small subset which is available in the derived table.

Listing 6.3: Second Optimization of Language Distribution Query

```sql
create index on fbdata_language_distribution (source);
create index on fbdata_language_distribution (date, source);
create index on fbdata_language_distribution (date);
```

Further optimization potential can be realized through the creation of indexes as depicted in Listing 6.3. Indexes normally created for all commonly queried fields in the table. In the present case, an index is created for each possible combination of the two table fields source and date.

After creation of the indexs on the derived table, we can measure that the performance of the SQL query depicted in Listing 6.2 has - again - improved.

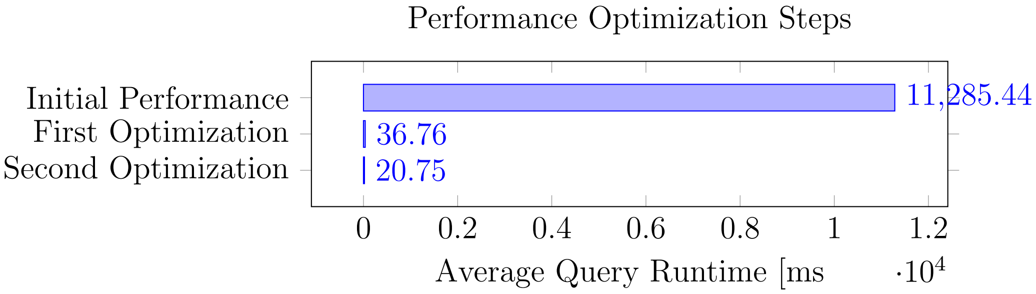


Figure 6.2.: Three-Step Performance Optimization of the Language Distribution Visualization

Figure 6.2 displays the results for each of the three different versions of the query for the Language Distribution visualization at hand. As explained above, the average query runtime is based on 10 data points that were measured for each query.

Hence, when comparing the performance of the initial query to the first optimization step, a speed improvement of 307 times ($11285.44ms/36.76ms = 307$) becomes apparent. Furthermore, the second optimization step, which creates indexes for the derived table, adds another factor 1.77 ($36.76ms/20.75ms = 1.77$) performance improvement. Overall, it is shown that the original query was improved by a factor of 543. In the example query, this resulted in a 10 second speed improvement.

Based on the database testing, we can perform query optimization measures such as in the example depicted above. In the presented example, the optimization measures significantly increased the performance of the Language Distribution Visualization.

6.2.2. Multi-Variable Query Optimization

Although the query optimization measures described in the previous section are generally applicable, situations exist in which very complex database queries need to be analyzed. Complex database queries are characterized by having a significant number of sub-queries, and contain difficult database operations.

Such a situation arose, after reaching feature completeness with the generation of the timeframe-based Actor Overlap visualization. In section 4.5.2, the software architecture of the *VennDiagram* component which depicts the Actor Overlap visualization is highlighted. During evaluation of the overall dashboard performance in a production environment, it became apparent that *VennDiagram* needed much longer to load than all other visualizations combined.

Example: Optimization of Actor Overlap by Timeframe visualization

Challenged by this observation, a multi-variable query optimization was performed. A thorough analysis of the initially used query presented several variable components within the database queries. These variable components can be expressed differently while, in the end, still returning the same result. Based on a combination of these variables, a faster performance of the query might be achievable. This possibility needs to be examined in a structured way.

In this example, four variables have been identified to affect overall query performance:

- **Storage of Temporary Query Results**: The query at hand calculates multiple sub-queries and performs extensive comparisons based on subquery results. During this process, temporary data can either be stored using a *Temporary Table* or *In-Memory.* Storage in temporary tables can be achieved by using the SQL `WITH` statement.
- **Enforcing of Uniqueness Constraint**: The Actor Overlap visualization needs unique sets of Actors in order to calculate correct overlap data. In SQL, the uniqueness of a set can be achieved by either using `DISTINCT` or `GROUP BY` statements.
- **Counting of Actors**: The counting of Actors within any of the sets depicted above can be performed either using the SQL functions `COUNT(*)` or `COUNT(actorid)`.
- **Overlap Calculation**: The intersections between sets of Actors, which are later displayed in the Venn diagram, can be calculated by using either `INTERSECT` or `INTERSECT ALL`. The latter only works with already unique sets of actors.

- **Sorting of (temporary) Results**: It is not clear whether sorting of Actor sets displays an impact on query performance, therefore an SQL `ORDER BY` clause needs to be tested in different configurations in order to evaluate its effectiveness.

	Test Query #:	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Category:	Query Variables:							
Storage	**Temp Table**		✓	✓	✓	✓	✓	✓
	In-Memory	✓						
Uniqueness	**DISTINCT**				✓		✓	
	GROUP BY	✓	✓	✓		✓		✓
Counting	**COUNT(*)**	✓	✓	✓	✓		✓	
	COUNT(actorid)					✓		✓
Overlap	**INTERSECT**		✓	✓		✓		
	INTERSECT ALL	✓			✓		✓	✓
	ORDER BY	✓	✓				✓	✓
	Avg. Execution Time [sec]:	73.40	81.62	89.69	88.26	78.61	**42.31**	48.73

Table 6.1.: Empirical Multi-Variable Evaluation of Database Query Performance during calculation of Actor Overlap Visualizations

Table 6.1 illustrates the process of multi-variable database query optimization during realization of the Visual Analytics dashboard.

Methodology For evaluation purposes, several steps of variable changes are performed. Each configuration of the complex query is then executed for $n = 10$ times, while the query execution time is measured. Subsequently, the average query execution time is calculated and taken into account to compare the configurations.

In this example, a total of seven different configurations of the variables introduced previously are tested. The initial query returned one row with seven fields in a total of 73.40 seconds. The outcome of this query optimization showcases that the sixth query configuration produced the best results. This result is also depicted by Table 6.1.

Hence, the optimal variable composition is to utilize *Temporary Tables* for subquery result storage, the `GROUP BY` statement for uniqueness, the `COUNT(*)` statement for counting, the `INTERSECT ALL` statement for overlap calculations, and sort the result using `ORDER BY`.

The optimized database query for calculation of the time-based Actor Overlap visualization is depicted in the following Listing 6.4.

Listing 6.4: Query for time-based Actor Overlap Visualization that has been optimized using Multi-Variable Query Optimization

```sql
WITH _before AS (SELECT DISTINCT actorid FROM _fb_actors_expanded WHERE "date" BETWEEN '2009-12-31' AND '2010-01-28'),
_during AS (SELECT DISTINCT actorid FROM _fb_actors_expanded WHERE "date" BETWEEN '2010-01-28' AND '2010-09-26'),
_after AS (SELECT DISTINCT actorid FROM _fb_actors_expanded WHERE "date" BETWEEN '2010-09-26' AND '2010-10-25'),
_before_after AS (TABLE _before INTERSECT ALL TABLE _after),
_during_before AS (TABLE _during INTERSECT ALL TABLE _before),
_during_after AS (TABLE _during INTERSECT ALL TABLE _after),
_all AS (TABLE _during_before INTERSECT ALL TABLE _during_after INTERSECT ALL TABLE _before_after)
SELECT
(SELECT count(*) FROM _before) AS actors_before,
(SELECT count(*) FROM _during) AS actors_during,
(SELECT count(*) FROM _after) AS actors_after,
(SELECT count(*) FROM _before_after) AS intersect_before_after,
(SELECT count(*) FROM _during_before) AS intersect_before_during
,(SELECT count(*) FROM _during_after) AS intersect_during_after
,(SELECT count(*) FROM _all) AS intersect_all;
```

To conclude this evaluational section on database testing, it was shown that database testing is able to improve overall application performance of the Visual Analytics dashboard through different optimization methods. On top of that, it highlights areas in the software where optimization potential exists. It is then database performance engineering which is needed to realize the hidden optimization potential.

6.3. User Testing

The ultimate way of dashboard evaluation presents user testing. Only when testing with humans, true strengths and weaknesses of a user interface become apparent. In previous chapters, it was established that an important non-functional requirement of the Visual Analytics dashboard presents a seamless user experience. For user testing, various methods such as traditional interviews and task-based user studies can be applied.

Based on above-stated requirements, the Visual Analytics dashboard presented in this thesis is evaluated with users in two different ways.

First, a task-based user study is performed in order to gain valuable insight into the value of the tool in realistic usage scenarios. Secondly, group interviews and debriefings are conducted to learn more about the quality of the tool through users' opinions.

6.3.1. Task-Based User Study

In order to assess the ease of use of the developed Visual Analytics dashboard, a task-based user study has been performed with $n = 5$ participants. All five participants present above-average education levels, a certain media affinity, and familiarity with using digital services.

Methodology During the task-based user study, each test subject is placed in front of a computer of their choice and presented with six tasks shown in Appendix A. The six tasks are presented to the test subjects one after the other, and each subject is encouraged to find an answer to the task at hand by using the newly-developed Visual Analytics dashboard. Furthermore, the six tasks are divided into three categories “Find something”, “Identify a Trend” and “Compare X to Y”, where each category is represented by two tasks.

The test subjects' progress in working on the task is measured and monitored as soon as they begin to use the Visual Analytics dashboard. The time needed until the subject presents a sufficient answer to the task at hand is measured. This measurement depicts the *task completion time.* The measurement is performed openly and with knowledge of the test subject.

Although an extensive end user documentation of the Visual Analytics dashboard has been prepared (see Appendix B), **no briefing on usage of the dashboard is performed** with the test subjects. The test subjects start right away working on the presented tasks.

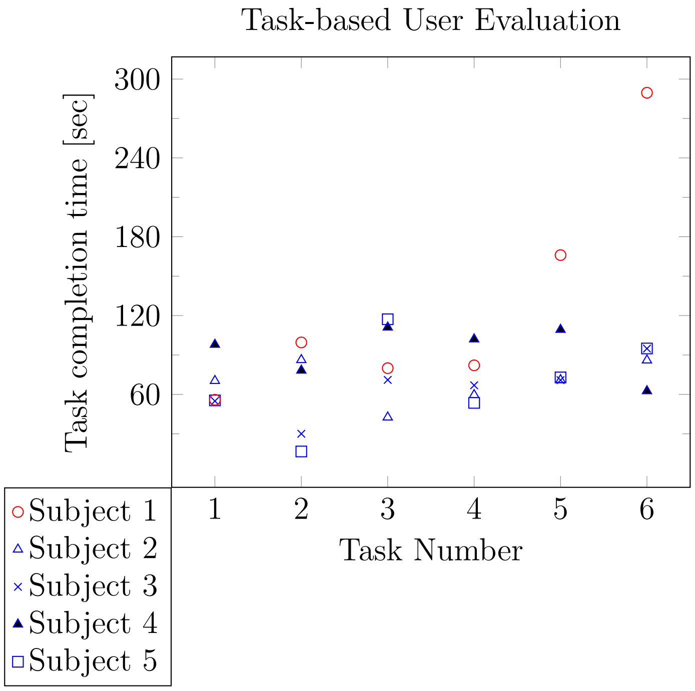


Figure 6.3.: Results of the Task-based User Study performed on the Visual Analytics Dashboard

Results of User Study Figure 6.3 showcases the results of the task-based user study for the five test subjects.

Subject 1 was the first performing the test. During subject's 1 test, previously unknown server-side problems occurred when working on tasks number five and six.

Subjects 2 to 5 worked with a trivially improved version of the Visual Analytics dashboard, which showed less performance decay over time, thereby allowing for a faster working with the software.

The average total task completion time for all subjects is 8 minutes and 29 seconds, whereas the average completion time per question is 1 minute and 25 seconds.

6.3.2. Peer Group Interviews

After finishing the user study, a debriefing of the test subjects was performed to find out about their sentiment with regard to the achieved goals and potential problems that occurred to them. During this debriefing, the examiner did not present the metrics to the test subjects, but rather asks them about input on further improvements of the dashboard.

Several test subjects exclaimed that they had fun during the study, and felt successful after being able to solve the tasks without further assistance. Some concerns were made with regard to the loading times of the *Actor Moblity* visualizations, in particular the Venn diagram, and the anxiety felt by test subjects in these moments.

All test subjects expressed the urge to use the Visual Analytics tool to find social media activity performed by one of their peers. After completion of the presented task, some subjects used the raw search functionality to search for some of their peers.

Furthermore, the test subjects presented new-found curiosity with regard to the disasters in the Bangladesh textile industry and asked the examiner for more information on these issues.

6.4. Limitations

Results of the evaluation have been presented in previous sections. Therefore, limits of the evaluation need to be discussed.

The user-based evaluation cannot be seen as highly credible due to the rather small sample size of five test subjects. On top of that, the sample size might be biased because all test subjects represented the same class of population.

The software-based evaluations have presented meaningful results from a small subset of components. The number of examined components needs to be increased in order to be able to create broader claims about the Visual Analysis dashboard.

Additionally, the examination of social media activity was reduced to eleven clothing retailers which might not give a thorough overview over all players in the garment industry. Most notably, many luxury brands were not included into the Visual Analytics tool. This might create a bias in the social media activity presented to the dashboard users, and thereby omit otherwise valuable insights into social media reactions to the Bangladesh events.

7. Related Work

Related work in the field of Big Data Visual Analytics in connection with event timelines can be found in various studies by Kaufhold and Reuter (2014), who are using post-event social media analysis of major disasters. One of these disasters was the 2013 flooding in Germany and its impact on spontaneous self-organization of emergency help on social media outlets such as Facebook and Twitter.

Other related research is depicted by further self-developed dashboards for analysis of social media data. This includes the IT artifact presented by Diakopoulos et al. (2010), which permits journalistic research in social media data from Twitter in relation to events of journalistic importance.

To conclude the listing of related work, Vatrapu et al. (2013) have performed various user studies on visual interfaces using eye-tracking approaches to gain actionable evaluation results.

The presented approaches are relevant for methodical improvements of the user study carried out in this thesis, which evaluated the newly developed Visual Analytics dashboard.

8. Summary & Conclusion

In this thesis, a detailed examination of contemporary challenges in Big Data Analytics has been performed. Research has reached a point where social media activity is ubiquitous, yet hard to collect and analyze. In conjunction with complex event timelines as depicted by the Bangladesh garment factory disasters, the data at hand presents numerous opportunities for attaining deep insights. In this context, visual analytics present the means of reaching those insights to many users with different backgrounds, both experts and novices alike.

The novel implementation of a Big Data Visual Analytics Dashboard designed and developed in the course of this thesis showcases, that the creation of visual analytics software which meets the high requirements of present-day datasets is viable, and can be achieved by a single programmer with limited resources. Furthermore, the developed IT artifact leverages open-source visual analytics frameworks to a maximum extent in order to achieve a pure implementation of important concepts in visual analytics such as the *detail on demand* principle.

A thorough evaluation showcased the effectiveness of the tool's approach on visual analytics. Both the client- and server-side components of the Visual Analytics Dashboard present performance at par with commercial tools, and can seamlessly be used under many operational circumstances.

Additionally, the results of the user study performed during this thesis indicate, that the presented Visual Analytics dashboard combines a high ease of use with the ability of performing many different interactive analyses on a large dataset. Moreover, the Visual Analytics tool put forward may be utilized through any modern browser on a multitude of different devices and screen sizes, with visualization display times as low as in the hundreds of milliseconds.

Complementing benchmarks of the database optimizations, which are applied to the Visual Analytics dashboard in real-world deployments, showcase good performance and hugely satisfactory handling of large amounts of social data. It was demonstrated that the visualizations depicted in the dashboard are accessible to novice users and relevant investigative user tasks were successfully solved. Hence, the presented Big Data Visual Analytics tool displays a pioneering approach and empowers further analysis efforts in the realm of social media data.

8.1. Outlook

This thesis presented a custom-tailored Visual Analytics tool for Big Social Data collected from Facebook. As Facebook depicts only one of many sources of social media activity, the focus of the tool needs to be expanded to incorporate other sources of social data.

Furthermore, the collection of social media activity should be more streamlined and a deeper integration with the SODATO tool needs to be targeted.

8.2. Future Work

Additional Features The implementation of further enhancements to the Visual Analytics dashboard needs to be performed. This mainly includes further data-driven UI and UX improvements which are validated by A/B-testing, and the implementation of additional visualizations such as a graph-based visualization of actor mobility.

Better Evaluation of the Visual Analytics Tool A larger user study with representative n needs to be performed. Furthermore, future studies need to use better equipment such as eye tracking appliances in order to facilitate more meaningful evaluation results. In these evaluations, a briefing could be performed in which the user manual is used. Then, the effect depicted by the user manual on task completion times can be more thoroughly analyzed.

Application of the Visual Analytics Dashboard to further research and business areas This includes changing the tool to work for social media marketing campaign monitoring and industry-specific monitoring purposes. Both of these situations include complex chains of events and strive for better visualizations of their impact in terms of social media activity.

Another application of the concept of the Visual Analytics Dashboard implemented in this thesis is the monitoring of political sentiment and conversation topics within the social media world of a large audience. This might for example be done with the Facebook walls of large political parties in European countries.

From this data corpus, researchers of political and social sciences gain yet another indicator for public opinion on current topics. On top of that, even some sort of shitstorm analysis dashboard or early warning system for massive waves of negative sentiment towards political actors could be realized.

Application to new Domains of complex Event Timelines Further work is not limited to social media marketing campaigns, but might enable further analysis-based fields such as forensic investigations to leverage the provided visual analytics tools. These professions mainly work on a fixed corpus of data with very large dimensions.

Based on the idea of expanding the current implementation of the detail on demand concept, a tool for analyzing complex forensic event timelines might be created based on the current framework.

This might enable the creation of a new set of investigative tools for (social) big data research from various data sources, and depict the online equivalent to established tools such as Jigsaw (Kang et al.; 2011).

Bibliography

Abousalh-Neto, N. A. and Kazgan, S. (2012). Big data exploration through visual analytics, Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on, pp. 285–286.

Abras, C., Maloney-Krichmar, D. and Preece, J. (2004). User-centered design, Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications **37**(4): 445–56.

Bajaj, V. (2012). Fatal fire in bangladesh highlights the dangers facing garment workers, New York Times **25**.

Bhavnani, S. K., Dang, B. and Divekar, R. (2013). Accelerating translational insights through visual analytics, AMIA.

Bostock, M. (2012). D3. js, Data Driven Documents .
URL: *http://d3js.org/*

Bravo-Marquez, F., Mendoza, M. and Poblete, B. (2014). Meta-level sentiment models for big social data analysis, Knowledge-Based Systems .
URL: *http://www.sciencedirect.com/science/article/pii/S0950705114002068*

Buhl, H., Röglinger, M., Moser, F. and Heidemann, J. (2013). Big data, WIRTSCHAFTSINFORMATIK **55**(2): 63–68.
URL: *http://dx.doi.org/10.1007/s11576-013-0350-x*

Burke, J. (2013). Bangladeshi factory collapse leaves trail of shattered lives, The Guardian .

Dahl, R. (2012). Node. js: Evented i/o for v8 javascript.
URL: *https://www.nodejs.org/*

Diakopoulos, N., Naaman, M. and Kivran-Swaine, F. (2010). Diamonds in the rough: Social media visual analytics for journalistic inquiry, Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on, pp. 115–122.

Gerasch, A., Faber, D., Küntzer, J., Niermann, P., Kohlbacher, O., Lenhof, H.-P. and Kaufmann, M. (2014). Bina: A visual analytics tool for biological network data, PLoS ONE **9**(2): e87397.
URL: *http://dx.doi.org/10.1371/journal.pone.0087397*

Haggerty, J. and Haggerty, S. (2009). Visual analytics of an eighteenth-century business network, Enterprise and Society .
URL: *http://es.oxfordjournals.org/content/early/2009/09/21/es.khp051.short*

Hearst, M. and Rosner, D. (2008). Tag clouds: Data analysis tool or social signaller?, Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, pp. 160–160.

Himi, S. A. and Rahman, A. (2013). Workers unrest in garment industries in bangladesh: An exploratory study, Journal of Organization and Human Behaviour **2**(3): 49–55.

Hussain, A. and Vatrapu, R. (2014). Social data analytics tool (sodato), Advancing the Impact of Design Science: Moving from Theory to Practice, Springer International Publishing, pp. 368–372.

Ihrig, C. J. (2013). The express framework, Pro Node. js for Developers, Springer, pp. 189–204.

Islam, M. A., Deegan, C. et al. (2014). Social audits and multinational company supply chain: A study of rituals of social audits in the bangladesh garment industry, Available at SSRN 2466129 .

Kaczmirek, L., Mayr, P., Vatrapu, R., Bleier, A., Blumenberg, M., Gummer, T., Hussain, A., Kinder-Kurlanda, K., Manshaei, K., Thamm, M. et al. (2013). Social media monitoring of the campaigns for the 2013 german bundestag elections on facebook and twitter, arXiv preprint arXiv:1312.4476 .

Kang, Y.-a., Gorg, C. and Stasko, J. (2011). How can visual analytics assist investigative analysis? design implications from an evaluation, Visualization and Computer Graphics, IEEE Transactions on **17**(5): 570–583.

Kaufhold, M.-A. and Reuter, C. (2014). Vernetzte selbsthilfe in sozialen medien am beispiel des hochwassers 2013/linked self-help in social media using the example of the floods 2013 in germany, i-com **13**(1): 20–28.

Keim, D., Kohlhammer, J., Ellis, G. and Mansmann, F. (2010). Mastering the Information Age - Solving Problems with Visual Analytics, Eurographics Association.
URL: *http://books.google.de/books?id=vdv5wZM8ioIC*

Key, A., Howe, B., Perry, D. and Aragon, C. (2012). Vizdeck: Self-organizing dashboards for visual analytics, Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12, ACM, New York, NY, USA, pp. 681–684.
URL: *http://doi.acm.org/10.1145/2213836.2213931*

Khanna, P. (2011). Making labour voices heard during an industrial crisis: workers' struggles in the bangladesh garment industry, TRAVAIL, capital et société **44**(2).

Lindesay, F. (2012). Jade - node.js template engine, Jade Template Engine .
URL: *http://jade-lang.com/*

Manik, J. A., Yardley, J. and DHAKA, B. (2013). Building collapse in bangladesh leaves scores dead, NY TIMES (Apr. 24, 2013), http://www. nytimes. com/2013/04/25/world/asia/bangladesh-buildingcollapse. html .

Merkle, F., Schäfer, H. and Zillessen, S. (2013). Evaluation verfügbarer visual analytics toolkits anhand von benchmark-datensätzen.
URL: *http://elib.uni-stuttgart.de/opus/volltexte/2013/8389*

Miller, H. and Mork, P. (2013). From data to decisions: A value chain for big data, IT Professional **15**(1): 57–59.

Otto, M. and Thornton, J. (2010). Bootstrap, Twitter Bootstrap .
URL: *http://getbootstrap.com/*

Rahman, S. (2013). Broken Promises of Globalization: The Case of the Bangladesh Garment Industry, Lexington Books.

SAS Institute Inc., S. (2014). Sas big data analytics software.
URL: *http://www.sas.com/*

Sato, H. (2014). Cournot competition and reduction of corruption to prevent garment factory fires in bangladesh.

Schatz, M., Phillippy, A., Shneiderman, B. and Salzberg, S. (2007). Hawkeye: an interactive visual analytics tool for genome assemblies, Genome Biology **8**(3): R34.
URL: *http://genomebiology.com/2007/8/3/R34*

Schlueter, I. (n.d.). The node package manager and registry.
URL: *https://www.npmjs.org/*

Stewart, K. L. (2013). An ethical analysis of the high cost of low-priced clothing, NABET p. 128.

Tableau Software Inc, T. (2014). Tableau visual analytics tool.
URL: *http://www.tableausoftware.com/*

Thomas, J. J. and Cook, K. A. (2005). Illuminating the Path: The Research and Development Agenda for Visual Analytics, National Visualization and Analytics Ctr.
URL: *http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0769523234*

TIBCO Software Inc, S. (2014). Tibco spotfire business intelligence analytics software & data visualization.
URL: *http://spotfire.tibco.com/*

Vatrapu, R., Reimann, P., Bull, S. and Johnson, M. (2013). An eye-tracking study of notational, informational, and emotional aspects of learning analytics representations, Proceedings of the Third International Conference on Learning Analytics and Knowledge, LAK '13, ACM, New York, NY, USA, pp. 125–134.
URL: *http://doi.acm.org/10.1145/2460296.2460321*

White, T. (2009). Hadoop: The Definitive Guide, 1st edn, O'Reilly Media, Inc.

Wong, P. C., Shen, H.-W., Johnson, C., Chen, C. and Ross, R. B. (2012). The top 10 challenges in extreme-scale visual analytics, Computer Graphics and Applications, IEEE **32**(4): 63–67.

Wong, P. C. and Thomas, J. (2004). Visual analytics, IEEE Computer Graphics and Applications **24**(5): 20–21.

Appendices

A. User Tasks for Evaluation

B. Dashboard User Manual